Intelligent and Sustainable Control for Energy-Efficient Open Radio Access Networks

Xuanyu Liang

Doctor of Philosophy

University of York

School of Physics, Engineering and Technology

April 2026

Abstract

The rapid densification of next-generation wireless networks, particularly toward Sixth Generation (6G), has increased the energy consumption and carbon footprint of Radio Access Network (RAN), where Radio Units (RUs) are a major source of power usage. This thesis investigates intelligent control mechanisms for RU operation management in Open Radio Access Network (O-RAN) systems, focusing on energy efficiency, scalability, and sustainability.

First, an O-RAN-compliant control framework is developed through intelligent xApps deployed in the Near Real Time RIC (Near-RT RIC). Using a commercial RAN Intelligent Controller (RIC) simulator, the proposed approach dynamically adapts RU operational states and achieves up to 50% power reduction while maintaining Quality of Service (QoS).

The RU sleep control problem is then formulated as a Markov Decision Process (MDP), and a Deep Reinforcement Learning (DRL) solution based on Twin Delayed Deep Deterministic Policy Gradient (TD3) is proposed. By using continuous action spaces, the proposed method avoids the exponential growth of action combinations and achieves over 50% energy savings compared with the always-on baseline, outperforming Deep Q-Learning Network (DQN)-based methods by up to 6% with improved convergence stability.

To address scalability, a federated DRL framework aligned with the hierarchical O-RAN architecture is introduced. Local agents operate within Near-RT RICs, while a global model is coordinated by the Non Real Time RIC (Non-RT RIC). This approach achieves up to 43.75% faster convergence and reduces training energy consumption by 37.4%, while maintaining network-wide energy savings above 50%.

Finally, the thesis extends RU control toward sustainability by incorporating carbon awareness. A carbon-aware DRL framework is developed for heterogeneous energy sources, enabling differentiated activation of renewable- and grid-powered RUs. Results show up to 20% lower carbon emissions than heuristic baselines, while joint energy-carbon optimization achieves approximately 12%–28% lower emissions than energy-only approaches.

Acknowledgments

First and foremost, I sincerely thank my supervisor, Hamed Ahmadi. Over the past four years, he has provided me with patient guidance, continuous encouragement, and selfless support whenever I encountered difficulties. Under his mentorship, I have grown from a student with little understanding of research into someone capable of conducting independent scientific work. This transformation has been invaluable to me.

I would also like to thank my colleagues in the ISA and ITWIN groups. The discussions, collaborations, and exchange of ideas we shared have made this journey intellectually stimulating and far less solitary. My gratitude also goes to my friends in York, who supported me during moments of low motivation and uncertainty, helping me regain confidence and move forward.

I would like to take a moment to acknowledge myself as well. This four-year PhD journey has not only been an academic endeavor but also a profound period of personal growth. It has taught me how to think more critically and holistically, and how to remain resilient in the face of setbacks, unsatisfactory results, and uncertainty. I have learned the true meaning of perseverance and the importance of maintaining focus.

Beyond academia, I owe my deepest gratitude to my parents. They have always been my strongest support, respecting my decisions and providing unconditional encouragement, both emotionally and materially. Without them, I would not be where I am today.

I am especially grateful to my girlfriend, Jingxi Li, who has been the closest companion in my life. Her optimism, understanding, and constant encouragement have given me strength and comfort throughout this journey.

Finally, I would like to thank everyone who has, in one way or another, contributed to this long journey. It is your kindness and support that have collectively shaped my PhD experience. This thesis is not only mine, but also yours.

Xuanyu Liang

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Publications arising from this thesis are acknowledged in the following section, entitled *Related Publications*.

Related publications

Published work

1. **Liang, X.**, Wang, Q., Al-Tahmeesschi, A., Chetty, S.B., Grace, D. and Ahmadi, H., 2024. Energy consumption of machine learning enhanced open RAN: A comprehensive review. IEEE Access, 12, pp.81889-81910.

2. **Liang, X.**, Al-Tahmeesschi, A., Wang, Q., Chetty, S., Sun, C. and Ahmadi, H., 2024, July. Enhancing energy efficiency in O-RAN through intelligent xApps deployment. In 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE.

3. **Liang, X.**, Al-Tahmeesschi, A., Chetty, S. and Ahmadi, H., 2025. Green O-RAN Operation: a Modern ML-Driven Network Energy Consumption Optimisation. arXiv preprint arXiv:2512.07006.

4. **Liang, X.**, Al-Tahmeesschi, A., Chetty, S., Cavdar, C., Canberk, B. and Ahmadi, H., 2026. Scalable machine learning-based approaches for energy saving in densely deployed Open RAN. IEEE Transactions on Green Communications and Networking.

5. Wang, Q., Chetty, S., Al-Tahmeesschi, A., **Liang, X.**, Chu, Y. and Ahmadi, H., 2024, October. Energy saving in 6g o-ran using dqn-based xapp. In 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (pp. 01-06). IEEE.

6. Sun, C., Fontanesi, G., Chetty, S.B., **Liang, X.**, Canberk, B. and Ahmadi, H., 2024, July. Continuous transfer learning for UAV communication-aware trajectory design. In 2024 11th International Conference on

Wireless Networks and Mobile Communications (WINCOM) (pp. 1-7). IEEE.

List of Abbreviations

5G Fifth Generation

6G Sixth Generation

BBU Base Band Unit

BS Base Station

BSs Base Stations

IoT Internet of Things

LTE-A Long Term Evolution Advanced

MAC Medium Access Control

MDP Markov Decision Process

ML Machine Learning

mmWave millimeter Wave

MIMO Multiple Input Multiple Output

O-RAN Open Radio Access Network

QoS Quality of Service

RF Radio Frequency

RL Reinforcement Learning

RRH Remote Radio Head

RRC Radio Resource Control

RRU Remote Radio Unit

SDN Software Defined Network

SNR Signal-to-Noise Ratio

UE User Equipment

DRL Deep Reinforcement Learning

AI Artificial Intelligence

RAN Radio Access Network

RU Radio Unit

CU Central Unit

DU Distributed Unit

NR New Radio

gNBs Next Generation Node Bases

CP Control Plane

UP User Plane

FPGAs Field Programmable Gate Arrays

ASICs Application-specific Integrated Circuits

PHY-low lower level PHY layer processing

FFT Fast Fourier Transform

RRC Radio Resource Control

SDAP Service Data Adaptation Protocol

PDCP Packet Data Convergence Protocol

RLC Radio Link Control

RIC RAN Intelligent Controller

KPMs Key Performence Measurements

RT Real Time

SMO Service Management and Orchestration

UE User Equipment

API Application Programming Interface

DRL Deep Reinforcement Learning

SL Supervised Learning

RL Reinforcement Learning

LSTM Long Short-term Memory

MDP Markov Decision Process

LSTM Long Short-term Memory

TD Temporal Defence

DRL Deep Reinforcement Learning

DQN Deep Q-Learning Network

DDQN Double Deep Q-Learning Network

eMMB enhanced mobile broadband

URLLC ultra-reliable low-latency communication

QoS Quality of Service

NF Network Function

VFN Network Function Virtualization

NBC Navie Bayes Classifier

RAT Radio Access Technology

FML federated meta-learning

TS Traffic Steering

CQL Conservative Q-Learning

REM Random Ensemble Mixture

SCA Successive Convex Approximation

v-RAN Virtual RAN

RRH Remote Radio Head

NFV Network Function Virtualization

PA Power Amplifier

mmWave millimeter Wave

LOS Line of Sight

NLOS Non-Line of Sight

Near-RT RIC Near Real Time RIC

Non-RT RIC Non Real Time RIC

TD3 Twin Delayed Deep Deterministic Policy Gradient

RC Radio Card

UMa Urban Microcell path loss

PRB Physical Resource Block

API Application Programming Interface

DDPG Deep Deterministic Policy Gradient

TD Temporal Difference

RSRP Reference Signals Received Power

FL Federated Learning

UMi Urban Microcel

BN Batch Normalization

SBS Small Base Station

FRL Federated Reinforcement Learning

DQNSA DQN-Single Action

DQNMA DQN-Multiple Action

RIS Reconfigurable Intelligent Surface

PPO Proximal Policy Optimization

UDN Ultra Dense Network

SMDP Semi-Markov Decision Process

FDRL Federated Deep Reinforcement Learning

SDN Software Defined Network

SNR Signal-to-Noise Ratio

UE User Equipment

DRL Deep Reinforcement Learning

AI Artificial Intelligence

RAN Radio Access Network

RU Radio Unit

CU Central Unit

DU Distributed Unit

NR New Radio

gNBs Next Generation Node Bases

CP Control Plane

UP User Plane

FPGAs Field Programmable Gate Arrays

ASICs Application-specific Integrated Circuits

PHY-low lower level PHY layer processing

FFT Fast Fourier Transform

RRC Radio Resource Control

SDAP Service Data Adaptation Protocol

PDCP Packet Data Convergence Protocol

RLC Radio Link Control

RIC RAN Intelligent Controller

KPMs Key Performence Measurements

RT Real Time

SMO Service Management and Orchestration

UE User Equipment

API Application Programming Interface

DRL Deep Reinforcement Learning

SL Supervised Learning

RL Reinforcement Learning

LSTM Long Short-Term Memory

MDP Markov Decision Process

TD Temporal Defence

DRL Deep Reinforcement Learning

DQN Deep Q-Learning Network

DDQN Double Deep Q-Learning Network

eMMB enhanced Mobile Broadband

URLLC Ultra-Reliable Low-Latency Communication

QoS Quality of Service

NF Network Function

VFN Network Function Virtualization

NBC Navie Bayes Classifier

RAT Radio Access Technology

FML Federated Meta-Learning

TS Traffic Steering

CQL Conservative Q-Learning

REM Random Ensemble Mixture

SCA Successive Convex Approximation

RRH Remote Radio Head

NFV Network Function Virtualization

v-RAN Vritualized-RAN

PA Power Amplifier

EPM Edge Processing Module

CPM Central Processing Module

CA Carrier Aggregation

CCs Carrier Components

MAC Medium Access Control

3GPP 3rd Generation Partnership Project

IFFT Inverse Fast Fourier Transform

CPU Central Processing Unit

MEC Multi-access Edge Computing

Contents

Abstract ii

Acknowledgments iii

Related publications v

List of Abbreviations vii

List of tables xx

List of figures xxi

1 Introduction 1
- 1.1 Background and Motivation . . . 1
- 1.2 Research Challenges and Objectives . . . 3
- 1.3 Structure and Main Contributions of the Thesis . . . 5

2 Literature Review 9
- 2.1 Overview of RAN Evolution . . . 10
  - 2.1.1 Distributed RAN (D-RAN) . . . 10
  - 2.1.2 Centralized RAN (C-RAN) . . . 11
  - 2.1.3 VirtualiZed RAN (v-RAN) . . . 11
- 2.2 Open RAN Key Innovations . . . 12
  - 2.2.1 Disaggregation . . . 13
  - 2.2.2 RIC and Closed-Loop Control . . . 15
  - 2.2.3 Virtualization . . . 18
  - 2.2.4 Open Interfaces . . . 19
- 2.3 Machine Learning for O-RAN optimisation . . . 20
  - 2.3.1 AI/ML workflow in O-RAN . . . 20
  - 2.3.2 ML-based Applications in O-RAN . . . 22

2.4 Energy Consumption in O-RAN 26
2.4.1 Energy Consumption Models of O-RU 28
2.4.2 Energy Consumption Models of DU and CU 31
2.5 BS Sleep Mode Control Techniques 34
2.5.1 Traditional Optimisation Methods 34
2.5.2 Centralized Learning-based Optimisation Methods . . . 36
2.5.3 Decentralized Learning-based Optimisation Methods . 37
2.6 Summary 38

3 Energy-Efficient xApp-Based Control in O-RAN 40
3.1 Introduction 40
3.2 System model 42
3.3 Problem Formulation 43
3.4 Radio Card (RC) Switching Algorithm 45
3.4.1 Proposed xApp1 46
3.4.2 Proposed xApp2 46
3.5 Results 49
3.6 Summary 53

4 Deep Reinforcement Learning for RU Sleep Control in O-RAN 54
4.1 Introduction 54
4.2 Foundations of Deep Reinforcement Learning for RU Sleep Control 56
4.2.1 Markov Decision Process Formulation 56
4.2.2 Value Functions and Bellman Equations 57
4.2.3 From Q-Learning to Deep Q-Networks 58
4.2.4 Limitations of DQN for RU Sleep Control 59
4.2.5 Policy-Based and Actor–Critic Reinforcement Learning 60
4.3 System Model 61
4.3.1 Power Consumption Model and Problem Formulation . 63
4.4 Energy Efficiency Deep Reinforcement Learning (DRL) based Solution 65
4.4.1 Markov Decision Process Problem 65
4.4.2 DQN-Based Model Training 69
4.4.3 TD3 Model Training 72
4.5 Simulation Results 74
4.5.1 Simulation Settings 74
4.5.2 Simulation Results 78
4.6 Summary 81

5 Federated Reinforcement Learning for Scalable O-RAN Optimization 83
5.1 Introduction . . . 83
5.2 Limitations of Centralized DRL . . . 85
5.3 Chapter Contributions . . . 86
5.4 System Extension for Multi-Region O-RAN . . . 87
5.4.1 Problem Formulation in Federated Setting . . . 88
5.5 Federated DRL for Scalable RU Control in O-RAN . . . 90
5.5.1 Federated DRL Training and Global Aggregation . . . 91
5.6 Simulation Results . . . 95
5.6.1 Simulation Settings . . . 96
5.6.2 Numerical Results . . . 98
5.7 Summary . . . 104

6 Carbon-Aware Reinforcement Learning for Sustainable O-RAN 105
6.1 Introduction . . . 105
6.2 Chapter Contributions . . . 107
6.3 System Model . . . 108
6.3.1 Power Consumption Model . . . 110
6.3.2 Energy Supply Model . . . 111
6.3.3 Carbon Emission Model . . . 112
6.3.4 Problem Formulation . . . 113
6.4 Sustainable DRL based solution . . . 115
6.4.1 Markov Decision Process Problem . . . 115
6.4.2 PPO-Based Training for RU Sleep Control . . . 118
6.5 Simulation Environment . . . 122
6.5.1 Simulation Settings . . . 122
6.5.2 Numerical Results . . . 128
6.6 Conclusion . . . 138

7 Conclusions and future work 140

Appendices 143

References 144

List of Tables

2.1 O-RAN-based ML applications 27

3.1 Simulation Parameters 51

4.1 Simulation Parameters 77
4.2 Network Configurations 77

5.1 Simulation Parameters 97

6.1 Simulation Parameters 124
6.2 Comparison of Reward Function Designs 125
6.3 Heuristic baseline parameters. 127

List of Figures

2.1 O-RAN Architecture, with components and interfaces from O-RAN and 3GPP. O-RAN interfaces are drawn as solid lines, 3GPP ones as dashed lines. And three different control loops. 13

2.2 Near-RT RIC internal architecture. 15

2.3 Non-RT RIC and SMO internal architecture. 16

2.4 O-RAN AI/ML General Procedure. 21

3.1 shows O-RAN Architecture, with components and interfaces from O-RAN and 3GPP. O-RAN interfaces are drawn as solid lines, 3GPP ones as dashed lines. The layout of the O-RU and the distribution of users is shown in the picture in the right section. Each O-RU consists of two RCs operating on distinct frequencies, n77 and n78, represented by red and green circles, respectively. Due to the identical coverage areas of both RCs, only the color of one RC is discernible in the figure. The color of the User Equipments (UEs) within the figure matches the RC to which they are connected, indicating their network association based on the respective RC's frequency. The proposed two xApps are anchored in Near-RT RIC 41

3.2 shows the flowgraph that how the python code based xApp interact with RIC-tester. 50

3.3 shows the power consumption in percentage among 10, 50 and 100 UEs in three different scenarios. All on is always in 100% power consumption. xApp1 and xApp2 show a relative percentage of the All on scenario. 50

3.4 shows the average (over 10 trials) number of RCs are switched into sleep mode between 10, 50 and 100 UEs in three different scenarios 52

4.1 shows O-RAN Architecture, with components and interfaces from O-RAN and 3GPP. O-RAN interfaces are drawn as solid lines, 3GPP ones as dashed lines. The layout of the O-RU is shown in the picture in the right section. 55

4.2 illustrates the radio maps of 500 × 500 m^2 and 1000× 1000 m^2 area. 75

4.3 The rewards of TD3, DQNSA and DQNMA in 500 × 500 m^2 scenario 78

4.4 The average energy consumption in 500 × 500 m^2 scenario among 10 to 40 UEs with DQNMA, DQNSA and TD3 models. 79

4.5 The rewards of TD3 and DQNSA model in 1000 × 1000 m^2 scenario 80

4.6 The average energy consumption among 20 to 80 UEs in 1000 × 1000 m^2 scenario with TD3 and DQNSA models. 81

5.1 Illustration of the O-RAN architecture incorporating both O-RAN-defined and 3GPP-standard interfaces. Solid lines represent O-RAN interfaces (e.g., E2, A1, O1, Open FH), while dashed lines indicate 3GPP interfaces.The system spans four geographical areas, each with its own O-DU and O-CU components. Near-RT RICs operate on a per-area basis, deploying multiple xApps. A centralized Non-RT RIC performs global policy aggregation and training coordination using A1 interface. 84

5.2 Illustration the workflow of Fed-TD3 across distributed agents and a aggregator. Red solid lines indicate local critic training based on temporal-difference loss; green dashed lines represent actor updates guided by critic gradients and soft update target network; black and purple solid lines denote the periodic aggregation and redistribution of actor and critic parameters via the aggregator. Each agent operates within its own local environment and contributes to a globally coordinated policy through federated learning. 92

5.3 The left plot shows a radio map of single 500 m×500 m area used for centralized training and evaluation. The right plot depicts a 1000 m×1000 m area composed of four such subregions, representing a composite environment for centralized training and inference. This comparison setting is used to evaluate the generalization capability of the global model trained via federated reinforcement learning versus a centralized model trained on the combined area. The middle plot presents a single large-scale 1000 m×1000 m environment used to evaluate model performance under centralized DRL training. 95

5.4 The rewards and convergence speed of TD3 and Fed-TD3 in 500 m×500 m area with 6 RUs and 20 UEs. 98

5.5 The rewards and convergence speed of Fed-TD3, Fed-DQNSA and Fed-DQNMA model in 500 m×500 m area with 6RUs and 20UEs. 99

5.6 Comparison of average energy consumption (right figure) and UE satisfaction (left figure) between federated and centralized reinforcement learning approaches. The first three bars represent federated models (Fed-DQN with multiple and single actions, and Fed-TD3) applied to four independent 500 m×500 m regions with 6 RUs and 20 UEs. The last two bars show the performance of centralized models (DQN and TD3) trained on a merged 1000 m×1000 m region with 24 RUs and 80 UEs. . . 101

5.7 The training energy consumption for different DRL models. Each curve represents the cumulative energy usage over time for a specific model. The red labels indicate the total energy consumed by each model upon convergence. The comparison highlights the efficiency differences among federated and centralized approaches. 102

6.1 Proposed heterogeneous energy network 108

6.2 Spatial deployment scenarios with different renewable RU penetration levels and layouts. Green circles indicate renewable-powered RUs, while gray circles represent grid-powered RUs. These scenarios are used to evaluate the robustness of RU sleep scheduling policies under varying renewable availability and spatial distributions. 123

6.3 Training convergence comparison between PPO and TD3 under the 6 renewable RU (random layout) scenario. The moving-average episode rewards are shown for 40 UE and 80 UE traffic loads, each evaluated with two independent reward functions (AG1 and AG2). 128
6.4 Average number of active renewable-powered and grid-powered RUs under different control strategies for various renewable penetration levels and spatial layouts. Each subfigure corresponds to a different deployment scenario. Results are shown for both 40 UEs and 80 UEs. 129
6.5 Average network energy consumption under different control strategies as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. Results are shown for TD3, PPO with two reward functions (AG1 and AG2) and heuristic baselines. 132
6.6 Average network energy consumption under different control strategies as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. Results are shown for TD3, PPO with two reward functions (AG1 and AG3) and heuristic baselines. 134
6.7 Average operational carbon emissions under different control strategies as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. The results illustrating how carbon-aware control policies increasingly rely on renewable-powered RUs and reduce grid-dependent emissions as renewable availability grows. 135
6.8 Average operational carbon emissions under different control strategies (AG1 and AG3) as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. 137

1 Introduction

1.1 Background and Motivation

The rapid growth of mobile data traffic and the increasing demand of ubiquitous wireless connectivity have significantly transformed the design and operation of cellular networks [1]. Emerging applications such as Internet of Things (IoT), immersive multimedia services, autonomous systems, and extended reality are expected to impose unprecedented requirements on wireless infrastructure in terms of data rate, latency and reliability. To meet these demands, modern cellular networks have undergone large-scale densification through the deployment of a substantial number of Base Stations (BSs) and small cells, particularly in urban environments [2]. While network densification significantly improves spectral efficiency and coverage, it also introduces considerable energy consumption. Several studies have shown that BSs account for approximately 70%-80% of the total energy consumption in mobile cellular networks [3]. A typical Fifth Generation (5G) macro BS consumes between 3.3 and 9 kW, with even higher values observed in massive Multiple Input Multiple Output (MIMO) and millimeter Wave (mmWave) deployments. As the wireless system evolves toward beyond-5G and the future 6G architecture, the number of radio access nodes deployed is expected to

increase dramatically, further intensifying the energy demand of network infrastructure. Consequently, improving the energy efficiency of Radio Access Network (RAN) has become a critical research problem for both economic and environmental reasons.

Traditional cellular networks are typically designed to support peak traffic conditions, which results in significant resource under-utilization during off-peak hours. In dense deployments, many BSs remain active even when traffic demand is low, leading to unnecessary energy consumption. Therefore, dynamically adapting network operation according to real-time traffic conditions has been widely recognized as an effective approach for improving energy efficiency in cellular networks[4][5]. One widely studied mechanism for achieving this objective is BS sleep mode control, where lightly loaded network nodes are temporarily switched to low-power states without violating user Quality of Service (QoS) requirements. Recent advances in Artificial Intelligence (AI) and Machine Learning (ML) have opened new opportunities for enabling intelligent and adaptive network management. In particular, DRL has emerged as a promising framework for solving complex control problems in wireless networks due to its ability to learn optimal policies through interactions with dynamic environments [6]. In parallel with the adoption of AI techniques, the Open Radio Access Network (O-RAN) architecture has recently emerged as a key technological paradigm enabling flexible and programmable wireless networks. Unlike conventional monolithic RAN architectures, O-RAN adopts a disaggregated structure that separates the BS functionalities into three major components: the Central Unit (CU), the Distributed Unit (DU), and the Radio Unit (RU) [7]. This architectural

transformation introduces standardized interfaces and virtualization capabilities that enable the integration of intelligent control mechanisms directly into network operations. A key component of the O-RAN architecture is the RAN Intelligent Controller (RIC), which provides a programmable platform for deploying advanced optimisation applications. The RIC framework includes Near Real Time RIC (Near-RT RIC) and Non Real Time RIC (Non-RT RIC) layers that support real-time control and long-term policy optimisation, respectively. Within this architecture, modular control applications known as xApps can be deployed to implement intelligent network functions [8].

Motivated by these developments, this thesis investigates intelligent control mechanisms for energy-efficient and sustainable operation of O-RAN using advanced ML techniques.

1.2 Research Challenges and Objectives

Despite the promising opportunities provided by AI-driven network optimisation and the O-RAN architecture, several key challenges remain in developing practical and scalable energy-efficient control frameworks for future wireless networks.

First, the dynamic nature of wireless environments makes energy-efficient infrastructure management a complex sequential decision-making problem. Network conditions such as user mobility, traffic demands and channel quality may change rapidly over time. Therefore, effective control mechanisms must be capable of adapting to continuously evolving network states while maintaining strict QoS requirements for users. Designing learning-based control policies that can effectively capture these dynamics remains a significant

challenge.

Second, the scalability of AI-based control mechanisms becomes increasingly important as network deployments grow larger and more complex. In dense O-RAN deployments, a large number of RUs may operate simultaneously within the same coverage area. When each RU can independently switch between active and sleep states, the number of possible network configurations increases exponentially with the number of nodes. This phenomenon, commonly referred to as the action space explosion problem, significantly complicates the application of conventional DRL algorithms [9]. Furthermore, centralized ML frameworks typically require collecting large volumes of network data at a central server for training purposes. As wireless network become geographically distributed and heterogeneous, centralized training can introduce substantial communication overhead and computational bottlenecks. Scalable learning approaches are therefore required to enable scalable AI-driven control in large-scale networks.

Third, sustainability considerations are becoming increasingly important in design of next-generation wireless systems. While most existing studies focus on minimizing total network energy consumption, the carbon footprint of network operation depends heavily on the energy sources used to power network infrastructure. In hybrid renewable-grid deployments, different network modes may have significantly different carbon impacts depending on their energy supply. Consequently,energy-efficient control mechanisms should also incorporate carbon-awareness in order to achieve environmentally sustainable operation.

In order to address these challenges, the main objective of this thesis is to

develop intelligent, scalable and sustainable control mechanisms for O-RAN using advanced DRL techniques. Specifically, this research aims to: (i) develop programmable energy-efficient control mechanisms within the O-RAN architecture. (ii) design DRL algorithms capable of dynamically optimizing RU activation policies. (iii) enable scalable learning frameworks that support large-scale O-RAN deployments. (iv) incorporate carbon-awareness into network control decisions to support sustainable wireless systems.

1.3 Structure and Main Contributions of the Thesis

This thesis investigates intelligent control mechanisms for improving the energy efficiency, scalability, and sustainability of O-RAN. The overall research progression follows a clear technical evolution. It begins with the design of practical xApp-based energy control mechanisms within the O-RAN architecture, then advances to DRL for more adaptive RU sleep control, extends further to Federated Deep Reinforcement Learning (FDRL) for large-scale and geographically distributed deployments, and finally incorporates carbon-awareness into the control objective to support sustainable operation under hybrid renewable–grid energy supply. The thesis is organized accordingly.

Chapter 2 presents a comprehensive review of the O-RAN architecture and the role of ML in improving the energy efficiency of modern RAN. The chapter first outlines the evolution of RAN architectures, tracing the development from traditional distributed RAN to centralized, virtualized, and ultimately disaggregated O-RAN systems. It then introduces the key archi-

tectural components of O-RAN, including the CU, DU, and RU, together with the RIC framework and the roles of Near-RT and Non-RT RICs in enabling intelligent network control. In addition, the chapter reviews ML techniques applied to O-RAN optimisation and examines existing energy consumption models associated with both O-RAN components and AI/ML workloads as an update version of EARTH project model. Through this analysis, the chapter highlights the opportunities and challenges of integrating AI-driven intelligence into O-RAN and establishes the technical foundation for the research contributions developed in the subsequent chapters of this thesis.

Chapter 3 presents the first technical contribution of the thesis by investigating how energy-aware radio control can be practically embedded into the O-RAN architecture through xApps. The new knowledge contributed by this chapter is the demonstration that RU-level energy saving can be achieved through lightweight Near-RT RIC control logic using standard O-RAN measurements, without requiring complex optimisation or learning models. By designing and evaluating two xApp-based RC switching mechanisms, the chapter provides insight into how rule-based control can reduce unnecessary radio operation while preserving user QoS in a realistic O-RAN emulation environment.

Chapter 4 extends this work by studying RU sleep control as a sequential decision-making problem. The contribution of this chapter lies in showing how the formulation of the RU activation action space affects the scalability and learning stability of DRL-based energy saving. In particular, the chapter demonstrates that continuous-action TD3 can avoid the combinato-

rial growth associated with discrete RU switching actions and can provide more stable and energy-efficient control than DQN-based approaches under dynamic user and traffic conditions.

Chapter 5 addresses the scalability limitations of centralized learning in large O-RAN deployments. The new knowledge contributed by this chapter is the demonstration that federated reinforcement learning can exploit the hierarchical O-RAN control structure to distribute training across multiple Near-RT RIC domains while maintaining network-wide energy-saving performance. This provides insight into how learning-based RU control can be scaled geographically without relying on a single centralized learner or requiring all local data to be collected at one point.

Chapter 6 extends the thesis from energy-efficient control to sustainability-aware network operation. The scientific contribution of this chapter is the finding that incorporating carbon intensity into the reward design changes the learned RU activation behaviour, encouraging the network to favour renewable-powered infrastructure when possible. This chapter therefore provides new understanding of how carbon-aware reinforcement learning can reshape O-RAN control policies and how energy saving and emission reduction objectives interact under heterogeneous energy supply conditions.

The technical chapters of this thesis are closely connected to the author's first-authored publications. Chapter 2 is supported by the review paper *Energy consumption of machine learning enhanced open RAN: A comprehensive review*, which provides the background on O-RAN architecture, AI/ML-enabled control, and energy consumption modelling. Chapter 3 builds on the paper *Enhancing energy efficiency in O-RAN through intelligent xApps*

deployment, where practical xApp-based RU sleep control is investigated. Chapter 4 is related to the paper *Green O-RAN Operation: a Modern ML-Driven Network Energy Consumption Optimisation*, which develops learning-based energy-saving methods for O-RAN operation. Chapter 5 is supported by the paper *Scalable machine learning-based approaches for energy saving in densely deployed Open RAN*, which addresses scalable learning and energy saving in large-scale O-RAN deployments. Finally, Chapter 6 extends this research direction from energy-efficient operation to sustainability-aware control by incorporating both energy consumption and carbon emissions into the optimisation framework.

2

Literature Review

This chapter reviews the key technologies and research efforts related to intelligent and energy-efficient operation of O-RAN. The aim of this chapter is to provide a comprehensive overview of existing research that forms the technical foundation of this thesis and to identify the limitations of current approaches. In particular, this chapter focuses on three main aspects relevant to the research presented in this work: the evolution of RAN architectures, the emergence of the O-RAN paradigm and its programmable control framework, and recent studies on applying AI and ML techniques to improve network efficiency and automation. The discussion builds upon our previously published survey on machine learning-enabled energy-efficient O-RAN systems [1], where the integration of AI techniques with disaggregated RAN architectures is systematically reviewed. In this thesis, the literature is reorganized and expanded in order to highlight the specific research challenges addressed in the subsequent chapters, particularly those related to intelligent radio unit control, scalable learning frameworks, and sustainable network operation.

2.1 Overview of RAN Evolution

To understand the emergence of O-RAN and its suitability for intelligent network control, it is first necessary to review the evolution of radio access network architectures. The development from distributed to centralized and virtualized RAN reflects a broader shift toward greater flexibility, resource pooling, and programmability in wireless networks.

2.1.1 Distributed RAN (D-RAN)

In 2G networks, both baseband and radio signals were processed by the Base Stations (BSs), enabling digital voice transmission and basic data services for mobile devices. However, the evolution to 3G/4G networks introduced a new architecture called Distributed RAN (D-RAN) [10], which replaced the unified BS function of 2G RAN with separate components: the Base Band Unit (BBU) and the Remote Radio Unit (RRU) or Remote Radio Head (RRH). In D-RAN, each cell site is equipped with its own BBU, responsible for processing the baseband signals. The RRU, located at the cell site, connects to the mobile devices for receiving and transmitting signals under the control of the BBU. To ensure efficient communication between the BBU and RRU, a high-capacity fronthaul link, often implemented using technologies like optical fibers, is employed. This fronthaul link facilitates high data rates and low latency between the BBU and the RRU, enabling seamless communication.

2.1.2 Centralized RAN (C-RAN)

Unlike the traditional distributed cell site-based architecture, the C-RAN adopts a centralized approach where baseband processing and control protocol operations are consolidated in a BBU pool located at a central site. The C-RAN architecture offers several advantages, including significant cost and energy efficiency gains resulting from hardware consolidation [11]. The centralized approach enables cooperative radio resource management, leading to improved network performance and resource utilization. Additionally, C-RAN provides scalability and flexibility to accommodate fluctuations in network traffic and emerging technologies. However, the adoption of C-RAN also introduces challenges [12], particularly the requirement for high-capacity and low-latency fronthaul connections to ensure seamless communication between the RRHs and the centralized BBU pool [13]. These connections play a crucial role in maintaining the synchronization and timely delivery of data between the network elements. Overall, C-RAN represents a transformative network architecture that delivers cost savings, improved performance, and energy efficiency through centralized baseband processing and cooperative resource management. While it presents challenges related to fronthaul connectivity, C-RAN offers a promising solution for addressing the evolving needs of mobile networks.

2.1.3 VirtualiZed RAN (v-RAN)

To address the diverse requirements of 5G networks, the concept of Vritualized-RAN (v-RAN) has been proposed, incorporating key technologies such as Network Function Virtualization (NFV) and Software Defined Network (SDN)

[14]. NFV enables the decoupling of network functions from dedicated hardware, allowing them to be implemented as software applications on standard servers. SDN, on the other hand, separates the network control plane from the data plane, enabling centralized control and dynamic resource management. These technologies play a crucial role in the scalability, flexibility, and efficiency of V-RAN. In a V-RAN architecture [15, 16], the traditional BBU functions are virtualized and referred to as the V-BBU. These virtualized BBUs run as software on commodity servers within a data center environment. To ensure the necessary processing power, these servers are often organized in clusters, forming a cloud-based BBU pool or Cloud-RAN. The Cloud-RAN is connected to the cell sites (network edge) where the RRHs are located. The crucial link between the Cloud-RAN and the RRHs is established through Fiber Ethernet links. These fronthaul links are essential to support real-time processing requirements in mobile networks.

2.2 Open RAN Key Innovations

While earlier RAN architectures improved resource centralization and virtualization, they did not fully address the need for openness, programmability, and multi-vendor interoperability. O-RAN extends this architectural evolution by introducing a set of key innovations that collectively support intelligent and flexible network operation. This section reviews the main architectural features of O-RAN, focusing on disaggregation, the RIC-based control framework, virtualization, and open interfaces, all of which are central to the research developed in this thesis.

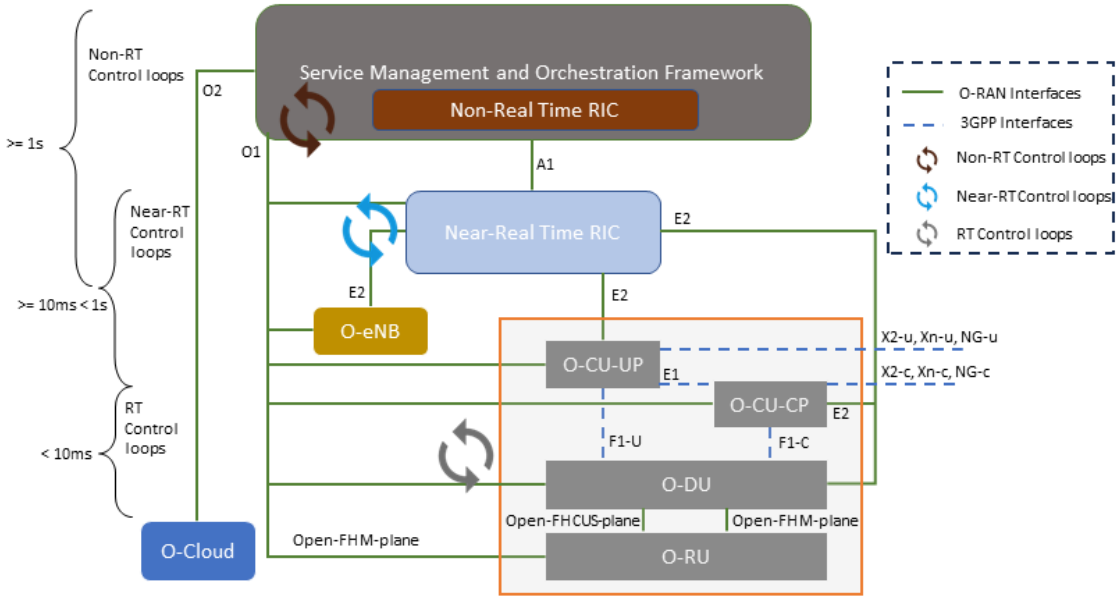


Figure 2.1: O-RAN Architecture, with components and interfaces from O-RAN and 3GPP. O-RAN interfaces are drawn as solid lines, 3GPP ones as dashed lines. And three different control loops.

2.2.1 Disaggregation

The first characteristic of RAN is disaggregation. The architecture of O-RAN is illustrated at Figure 2.1. It splits the BS into units with different functions. The structure corresponds to the New Radio (NR) Next Generation Node Bases (gNBs) defined by 3rd Generation Partnership Project (3GPP). The gNB is split into CU, DU and RU. where CU is further divided into two distinct parts, Control Plane (CP) and User Plane (UP). This splitting method allows each part to exist independently of the other. Different functions can therefore be located in different parts of the network or on different hardware platforms, making the whole network more flexible and versatile. The different functions of those three parts are as follows. The interfaces of them will be given in later section.

Open-Radio Unit (O-RU)

RUs, typically constructed using Field Programmable Gate Arrays (FPGAs) and Application-specific Integrated Circuits (ASICs) without virtualization, are situated in proximity to the Radio Frequency (RF) antennas, as highlighted by [17]. The O-RAN Alliance has undertaken comprehensive assessments of various 3GPP-endorsed RU and DU partitioning protocols, showing a particular inclination towards the 7.2x division [17], given it balances the time delay with inter-network traffic. This specific split entails a distribution of the physical layer across the RU and DU. In this arrangement, the RU manages solely the lower level PHY layer processing (PHY-low) functions, encompassing tasks such as Fast Fourier Transform (FFT)/ Inverse Fast Fourier Transform (IFFT) and cyclic prefix manipulation, aiming to mitigate deployment complexities and associated expenses. Beyond PHY-low operations, RU's architecture integrates RF processing elements, including power amplifiers, beamformers, and transceivers, to constitute a comprehensive operational unit.

Distributed Unit (O-DU)

The DU is responsible for the remaining higher level physical layer processing (PHY-high) such as channel modulation, part of precoding, and mapping into physical resource blocks, the Medium Access Control (MAC) layer and the Radio Link Control (RLC) layer. These three layers are generally closely synchronized due to the MAC layer generate Transport Blocks for physical layer by using the data buffered in RLC layer.

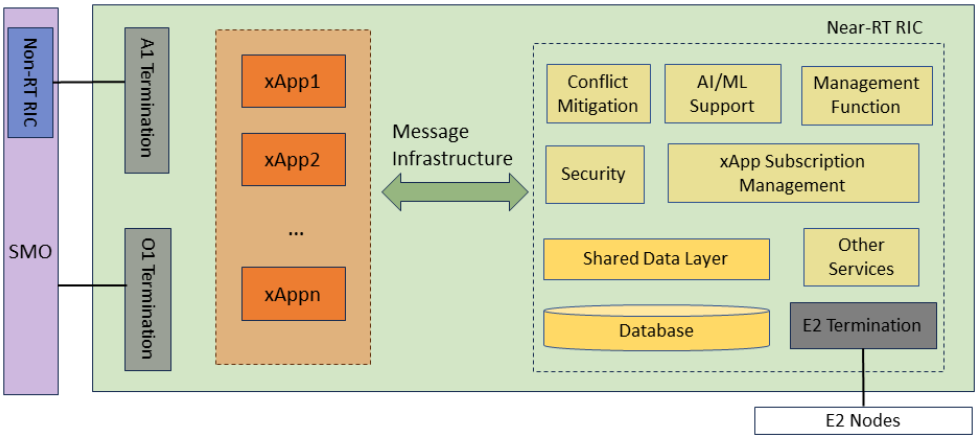


Figure 2.2: Near-RT RIC internal architecture.

Central Unit (O-CU)

The CU being placed higher up the 3GPP stack [18], which takes care of Radio Resource Control (RRC) layer, Packet Data Convergence Protocol (PDCP) layer and Service Data Adaptation Protocol (SDAP) layer. O-CU in O-RAN performs a wide range of radio functions, including efficient resource management, signal processing, connection management, quality of service optimisation, network slicing support, and interface coordination [19]. Its comprehensive role is vital for enhancing the performance, efficiency, and flexibility of the RAN within the O-RAN architecture.

2.2.2 RIC and Closed-Loop Control

Open RAN's notable second innovation is the RIC, a programmable entity designed to navigate the increasing complexities within network frameworks, thereby enhancing both accuracy and operational efficiency. By serving as a central network abstraction, the RIC compiles comprehensive Key Performence Measurements (KPMs) data, reflecting various facets of net-

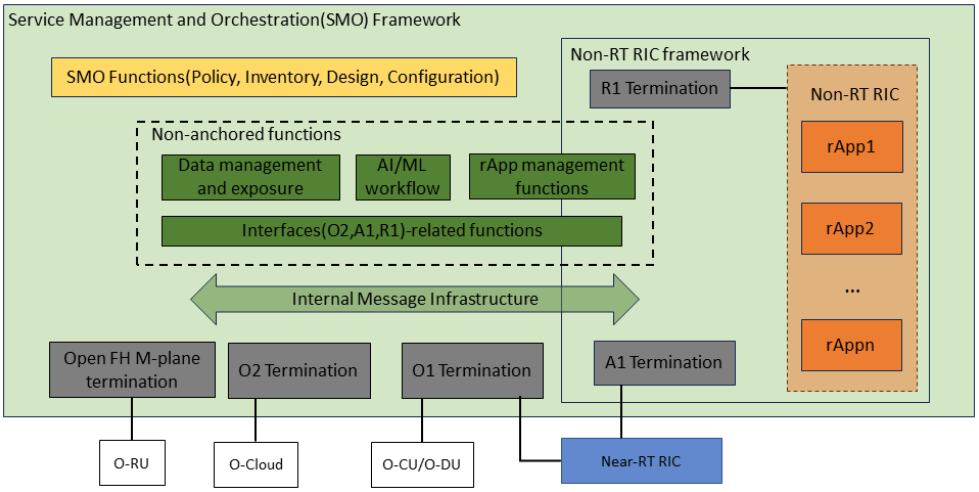


Figure 2.3: Non-RT RIC and SMO internal architecture.

work infrastructure status (for instance, user metrics, traffic burdens, and throughput capacities) and information from resources outside of RAN. This comprehensive data spectrum is analyzed through AI and ML methodologies, facilitating decision-making pertaining to RAN policies and operational strategies [20].

The non-Real Time (RT) RIC and near-RT RIC are the two primary modules of RIC that O-RAN Alliance has introduced [19] as illustrated at Figure 2.2 and Figure 2.3. The control loop with RAN components is carried out by a near-RT RIC with a time scale between 10ms and 1s, while a non-RT RIC is in charge of operations on a time scale larger than 1s, such as training AI and ML models as mentioned above. Figure 2.1 provides an overview of the closed-loop control implemented by the RICs across the disaggregated O-RAN infrastructure and the real-time extensions that need to be considered in the future. In the following paragraphs we will discuss the functionality of the different RICs and the closed-loop control associated with them.

Non-real-time RIC and Control Loop

The non-RT RIC constitutes an integral segment of the Service Management and Orchestration (SMO) framework, maintaining a direct connection with the A1 interface to near-RT RIC, as illustrated in Figure 2.1. Rather than operating through mutual interfacing within the SMO, the non-RT RIC exists as a specialized subset, executing only select functions of the broader SMO capabilities [8]. This component monitors operations related to RAN constituents on a temporal scale exceeding 1 second and facilitates AI/ML workflows, encompassing aspects like model training and configuration. Furthermore, it provides guidance and enrichment information to the Near-RT RIC utilizing a non-real-time control loop. Its connection, both direct and collateral, with interfaces like A1, O1, and O2 through the SMO, empowers the non-RT RIC to administer and configure RAN elements efficiently. An extensive discussion regarding the non-RT RIC and SMO is deferred to Section V.

Near-real-time RIC and Control Loop

The near-RT RIC, functioning as a specialized Open-RAN network function, orchestrates near real-time supervision and enhancement of E2 Node resources and services. This is achieved through fine-grained data gathering and subsequent operations executed via the E2 interface, with control loops operating within an expedited 10 ms to 1-second timeframe. The near-RT RIC consists of multiple applications to support network functions, called xApps. An xApp hosts multiple microservices such as mobility management, traffic steering, radio connection management, and QoS management.

Normally, an xApp receives the data from RAN components (i.e. CUs and DUs) and xApp sends back the action result after computing. At the same time, near-RT RIC also provides a range of functionality to support xApp.

2.2.3 Virtualization

The third characteristic of Open-RAN is the O-Cloud which is a cloud computing platform to manage and optimize the network infrastructure and operations, changing the network from edge system to virtualization platform. All the components mentioned in Figure 2.1, can be deployed in the O-Cloud Platform, which includes the following characteristics [21]:

- The O-Cloud platform combines hardware and software components that support cloud computing capabilities to execute RAN network functions.
- The cloud platform hardware is standardized to support the requirement of RAN functions to satisfy their performance objectives.
- The software in the O-Cloud platform exposes the open and well-defined Application Programming Interface (API) to manage and orchestrate life cycle of network functions and O-Cloud as well.
- The software is decoupled from the hardware, which means it is available from a variety of vendors.

The virtualization of O-RAN components and internal network functions plays a crucial role in energy conservation. Virtualization enables the flexible allocation of network resources based on user demands, optimizing the use

of network functions and, consequently, reducing energy consumption [22]. Furthermore, through the application of closed-loop control, as discussed earlier, dynamic cell activation and deactivation are facilitated, contributing to additional reductions in power consumption [23].

2.2.4 Open Interfaces

Lastly, the O-RAN Alliance in order to overcome the limitation of RAN has introduced well-defined specifications that uses open interfaces to connect different O-RAN elements. Figure 2.1 demonstrates the open interfaces defined by O-RAN and internal interfaces as defined by 3GPP. Open interfaces allow for diversity of options and operators can choose from a variety of vendors in different locations which breaks up vendor monopolies and increases market competitiveness.

As shown in Figure 2.1, F1 and E1 interfaces are defined from 3GPP. F1 interface builds the connection between DUs and CUs. E1 interface likes a bridge binding the user planes and control planes at CU. The rest interfaces are all defined by the O-RAN Alliance, the E2 interface connects near-RT RIC to RAN nodes. The DU and CU send measurements to the near-RT RIC through the E2 interface and then the configuration command back to the CU and DU to form a near-real-time control loop shown in Figure 2.1. The A1 interface [24] facilitating information exchange between the near-RT and non-RT RIC, which enables AI/ML related parameters or models deploy on the near-RT RIC. Besides, the non-RT RIC and SMO also connect to the O-Cloud via O2 interface for SMO to intelligently manage and operate each O-RAN node running on the top of the O-Cloud platform. Finally the O1

interface, which is standardized by the 3GPP [18], is used for management and optimisation of the RAN nodes.

The open interface characteristic of Open-RAN notably enhances its adaptability, permitting a flexible arrangement of components within the O-RAN architecture. This flexibility eliminates the need for a set location for network elements, enabling a configuration that is tailor-made to the specific demands of the network. For instance, the work in [25] positions the CUs and DUs at the network edge, with the RUs situated at the cell sites. Alternatively, there exists a scenario where the DUs and RUs are jointly situated at cell sites [26], leaving only the CUs at the edge, specifically to accommodate services sensitive to delays.

2.3 Machine Learning for O-RAN optimisation

The architectural flexibility of O-RAN is particularly significant because it enables intelligence to be embedded directly into network control loops. In this context, AI and ML are no longer peripheral tools for offline analysis, but become integral components of network optimisation and automation. This section therefore reviews how ML is incorporated into O-RAN.

2.3.1 AI/ML workflow in O-RAN

The integration of AI/ML into Open RAN architecture aims to transform RAN into more intelligent, efficient and adaptable system and even some of the theoretical ideas can be implemented in reality. By automating net-

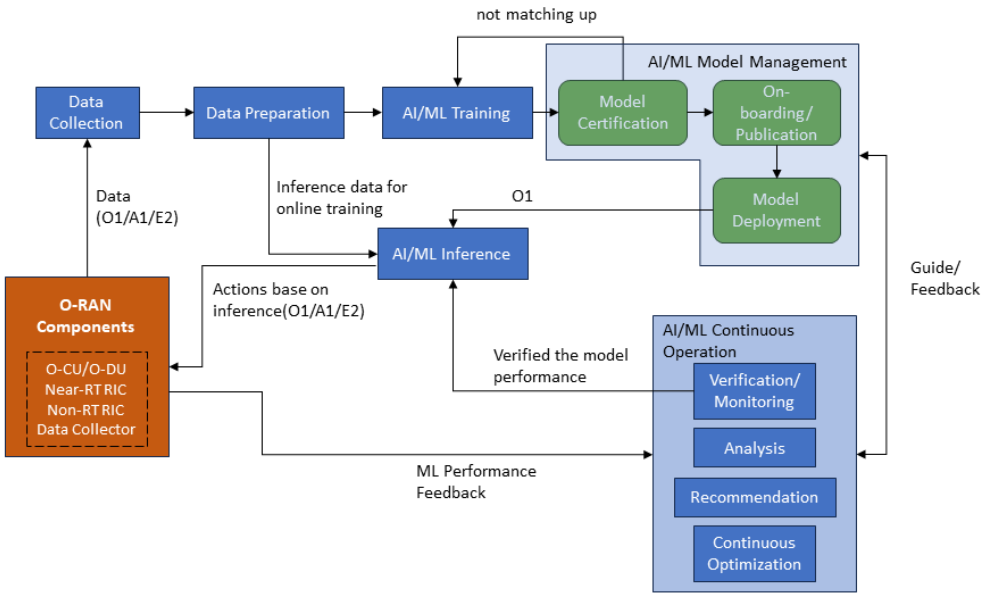


Figure 2.4: O-RAN AI/ML General Procedure.

work resource allocation, optimizing traffic steering, enhancing security, etc. AI/ML indeed facilitate management of increasingly complex wireless networks. This section is going to provide an overview of AI/ML procedure and AI/ML deployment scenarios, which are being standardized by O-RAN alliance in [27].

Figure 2.4 illustrates the AI/ML general workflow of O-RAN architecture. First, O-RAN infrastructure provides data through O-RAN interfaces (O1, A1 and E2) to the data collection block. This could be the network usage, throughput, latency, channel quality information and other relevant parameters, since different AI/ML solutions might require different data collection. The data is then structured and transformed as needed for specific used AI/ML models in the data preparation block. All AI/ML solutions that have collected data require to be trained before the deployment [27] while ensuring the accuracy and reliability to avoid outages or inefficiencies in the

network. Then trained models are going through the validation phase in AI/ML model management section to determine if the models perform as required when executing different network tasks. After that, the validated models are published in the SMO/Non-RT RIC catalogue. If the validation fails, they have to be re-trained or re-designed until pass the validation. The trained model in the AI/ML marketplace can be selected and deployed via containerized image to the execution node which refers to the inference host. Based on the output of the deployed ML model, actor in AI/ML assisted solution is informed to perform some control tasks including policy delivering (over A1 and E2 interfaces), configuration management (over O1 interface) and control action (over E2 interface). The last segment is AI/ML continuous operation, which has a crucial role in the overall workflow. It has the ability to detect and analyses intelligent models deployed throughout the network. In the event that models are not up to expectation in terms of reliability, accuracy or efficiency, it has the power to make those poorly performing models change their parameters and retrain them until they regain their original functionalities.

2.3.2 ML-based Applications in O-RAN

With the workflow in place, O-RAN enables ML techniques to be applied across a wide range of optimisation tasks. Rather than treating intelligence as a standalone add-on, these applications embed learning directly into operational decision-making, thereby improving network adaptability, efficiency, and automation.

Network Function relocation and DU-CU placement

The placement of DU and CU, along with the relocation of Network Function (NF), constitutes a critical aspect of optimizing the performance of specific applications within the O-RAN architecture. This optimisation is necessitated by the fact that different layers of the O-RAN architecture are designed to manage distinct operational objectives. For instance, CU, typically housed in data centers, possess superior computing capabilities compared to DU. This distinction renders CU more adept at processing complex network functions, while DUs are more suited to executing algorithms of relative simplicity. Consequently, the strategic allocation of network functions to their appropriate computational units, ensuring that each function is executed within the optimal environment, enhances the overall efficiency and performance of the network. This paper [28] considers the CU-DU placement and user association as a multi-objective optimisation problem. a novel DQN-based algorithm is proposed to reduce the cost of the O-RAN deployment and the end-to-end latency of UE by placing the DU-CU network functions in the regional and edge O-Cloud nodes, while jointly link users to RU. The aim of this paper is to enable RU to find the optimal O-Cloud node to run the network functions of DU and CU and the UE can also be assigned the most appropriate RU. In [29], authors add traffic type, delay budget and other requirements in to consideration. In this paper, two actor-critic learning based algorithm are proposed to decide resource allocation intelligently and relocate the resource allocation function dynamically at proper place (CU or DU). The aim of this paper is not only allocating the limited resources to the users but further improve the performance of NF by the proper CU-DU

selection. Based on [29] [30], the authors propose an upgraded version in [31]. In addition to the previous requirements, the agent in this paper needs to consider the propagation delay and energy consumption at the same time before choosing the proper location at DUs

Resource Allocation and O-RAN Slicing

O-RAN slicing is a significant application that offers more flexibility and precise network services, playing a crucial role in end-to-end network slicing. However, resource management, which involves allocating limited available resources to users while considering their QoS requirements and dynamic network situations, poses a challenging issue in RAN slicing [32]. In [20], an approach that combines a data-driven DRL-based algorithm deployed in as an xApp in the Near-RT RIC with the Colosseum open cellular stack. This integration enables the selection of the best-performing scheduling policy for each RAN slice, considering variations in the number of resource blocks allocated to each slice at different times, thereby ensuring enhanced scheduling accuracy. To address the allocation of communication and computational network resources for Ultra-Reliable Low-Latency Communication (URLLC) end devices, [33] proposes a two-level O-RAN slicing system. They treat the communication and computation resources as resource blocks from BSs and the Central Processing Unit (CPU) usage of Multi-access Edge Computing (MEC) servers. In [34], two different xApps are designed for power and resource allocation based on Reinforcement Learning (RL), along with a federated learning approach to coordinate the two xApps agents. This enables achieving higher maximum throughput for enhanced Mobile Broad-

band (eMMB) slices and lower latency for URLLC slices. [35] propose a policy-gradient based RL to address the allocation of computing resources. The objective is to maximize equitable allocation and optimize the QoS of the users, leading to superior performance compared to conventional methods. Similarly, [36] develops an edge network simulator based on the Q-Learning RL to demonstrate the performance improvement achieved by efficiently allocating resources. Overall, these studies contribute to the advancement of resource management in O-RAN slicing, utilizing RL algorithms and data-driven approaches to ensure more efficient and effective allocation of network resources.

Traffic Steering

In the intricate and multi-faceted ecosystem of an O-RAN, the Traffic Steering (TS) acts as an intelligent manager, responsible for optimizing complex data flows across diverse network elements while balancing the distribution of network resources to prevent overloading. Its primary objective is to enhance end-user experience by facilitating seamless and efficient data transmission. ML-based TS deployed at the near-RT RIC proves more suitable for adapting to evolving network conditions compared to traditional solutions, thanks to its centralized abstraction of the network that learns potential associations between different RAN parameters. In [37], a two-tiered ML algorithm combining Navie Bayes Classifier (NBC) and deep Q-learning is presented for traffic congestion prediction. This method employs Supervised Learning (SL) to predict congestion in each Network Function Virtualization (VFN) and feeds the output data to a DRL agent, which dynamically

steers traffic to avoid congestion, reduce queuing time, and ensure reliability. Similarly, in [38], an Long Short-Term Memory (LSTM) method and an Successive Convex Approximation (SCA)-based iterative algorithm are introduced for long and short sub-problem traffic prediction, respectively. Handover management, known as Radio Access Technology (RAT) allocation, is another significant use case of O-RAN, particularly crucial as the 5G system supports multiple access technologies, each with different access types. In [39], a Federated Meta-Learning (FML) algorithm is proposed for RAT allocation, significantly improving RL agents' training efficiency and adaptability to dynamic environments. This algorithm achieves a higher caching rate compared to a single RL agent while meeting UE demands. In comparison, [40] propose a comprehensive O-RAN-compliant framework for handover management and dual connectivity in the 3GPP network. To optimize the Traffic Scheduler, Conservative Q-Learning (CQL) and Random Ensemble Mixture (REM) techniques are employed, maximizing throughput and connecting O-RAN to the ns-3 simulation environment to obtain a vast training dataset, thus improving model accuracy.

2.4 Energy Consumption in O-RAN

Compared with conventional RAN architectures, the disaggregation of functions across RU, DU, and CU changes the way energy is consumed and distributed across hardware and software components. A clear understanding of these energy characteristics is essential for designing effective control strategies. For this reason, this section reviews existing models for characterising energy consumption in O-RAN, with particular emphasis on the RU

Table 2.1: O-RAN-based ML applications

Ref.	Application	AI/ML approach	Deployment	Advantages
[28]	CU-DU placement; user association	Deep Q-Network (DQN)	Non-RT RIC	Reduces deployment cost and UE end-to-end latency.
[29]	CU-DU selection; resource allocation	Actor-Critic (A2C)	Near-RT RIC	Improves NF performance.
[31]	CU-DU selection; energy efficiency	Actor-Critic (A2C)	Near-RT RIC	Improves energy efficiency and reduces mean latency.
[20]	RAN slicing; resource allocation	Deep Q-Network (DQN)	Near-RT RIC	Improves spectrum utilization and reduces buffer size.
[33]	RAN slicing; communication and computational resource allocation	Double Deep Q-Network (DDQN)	Near-RT RIC	Improves efficiency in meeting QoS requirements.
[34]	RAN slicing; power and resource allocation	Federated DRL	Near-RT RIC	Achieves higher throughput and lower latency.
[35]	Computing resource allocation	Policy-gradient-based RL	Near-RT RIC	Improves network performance.
[36]	Network resource allocation	Q-Learning	Near-RT RIC	Improves network performance.
[37]	Traffic prediction	Naive Bayes Classifier (NBC); Deep Q-Learning	Near-RT RIC	Reduces queuing time and traffic congestion.
[38]	Traffic prediction	Long Short-Term Memory (LSTM); Successive Convex Approximation (SCA)	Non-RT RIC	Provides highly accurate traffic prediction.
[39]	Handover management; dual connectivity	Federated Meta-Learning (FML)	Near-RT RIC	Achieves higher caching rates.
[40]	Handover management; dual connectivity	Conservative Q-Learning (CQL); Random Ensemble Mixture (REM)	Near-RT RIC	Improves throughput and spectral efficiency.

and the virtualized DU/CU components.

2.4.1 Energy Consumption Models of O-RU

EARTH Model of RU

The study in [41] defines power consumption model known as EARTH[1] model for a single BS that can be widely used. This is a typical BS consists of multiple transceivers, each of which serves one transmit antenna element. Therefore, power consumption of a BS can be regarded as the power consumption of all transceivers, each one including a Power Amplifier (PA), a RF module and a BBU, a DC-DC power supply, an active cooling system, and mains supply for connection to the electrical power grid. This BS power consumption is given by

$$P_{BS} = N_{TRX} * \frac{\frac{P_{out}}{\eta_{PA}\left(1-\sigma_{feed}\right)} + P_{BB} + P_{RF}}{\left(1-\sigma_{co}\right)\left(1-\sigma_{DC}\right)\left(1-\sigma_{MS}\right)} \tag{2.1}$$

where N_{TRX} denotes the total number of RF transceivers in a BS. P_{out} is the transmission power, scaling with traffic load, η_{PA} is the PA efficiency, σ_{feed} is the power losses by antenna feeder since the location of the macro BS is often different from its antenna, but this problem was addressed by having the RRH in the same site. P_{BB} and P_{RF} are the power consumption of BBU and RF module respectively. The power losses in active cooling system, DC-DC power supply and main power supply are denoting σ_{co}, σ_{DC} and σ_{MS}.

The simulation in [41] shown that only PA scales with BS load, other components hardly scale with the load. So the relation between BS power

[1]Energy Aware Radio and neTwork tecHnologies (EARTH) was an EU FP7 multi national project (2010-2012) which investigated the energy consumption of different network components.

consumption P_{in} and RF output power P_{out} are nearly linear and the equation can be simplified into the equation as follow:

$$P = \begin{cases} N_{TRX} * P_0 + \xi_P P_{out} & 0 < P_{out} < P_{max} \\ N_{TRX} * P_{sleep} & \mathrm{P}_{out} = 0 \end{cases} \tag{2.2}$$

where P_0 and ξ_P indicate the fixed power consumption and the slope of the load-dependent power consumption in different cell type, respectively. [41] argues that putting BS into sleep mode when its has no service to offer is critical to energy efficiency. Therefore, the $P_{sleep}\,(P_{sleep} < P_{out})$ indicates the power consumption when BS is entering sleep mode when there is no traffic to transmit

In the context of O-RAN, RU is a physical node[19] and a major power consumption of RU is related to RF functionalities and power amplification[42]. Therefore, the power consumption of RU can be formulated as follow base on the EARTH model:

$$P_{RU} = \begin{cases} P_{RF} + \frac{P_{out}}{\eta} & 0 < P_{out} < P_{max} \\ P_{sleep} & \mathrm{P}_{out} = 0 \end{cases} \tag{2.3}$$

where η is the power amplifier drain efficiency. P_{RF} is refer to fixed power consumption of RU such as RF circuits power consumption. P_{sleep} is the power consumption when RU is sleep.

Aggregation Power Consumption Model of RU

Carrier Aggregation (CA), introduced in the 3GPP Long Term Evolution Advanced (LTE-A) standard, serves as a pivotal feature to enhance cell throughput [43]. Fundamentally, CA permits the aggregation of multiple Carrier Components (CCs), thereby expanding bandwidth and augmenting data rates for mobile devices. This mechanism optimizes the use of the available spectrum by amalgamating carriers from either identical or disparate frequency bands. In its early stages, CA supported the aggregation of a maximum of 5 CCs with a bandwidth of up to 20 MHz. However, with the advent of 5G NR, its capabilities have been broadened to accommodate up to 16 CCs with a bandwidth reaching 1 GHz. This versatile technology has been instrumental in various areas, encompassing capacity augmentation, coverage enhancement, and facilitation of advanced functionalities such as Licensed Assisted Access (LAA) and dual connectivity [6].

Given this context, when constructing a power consumption model for RU utilizing CA, it becomes imperative to formulate a function that captures the power consumption's correlation with the number of active CCs, denoted as N_{cc}. Integrating insights from both the O-RAN framework and the study by [44], energy model for the RU is illustrated as follow:

$$P_{RU} = \sum_{j=1}^{N_{cc}} (P_{TX_j} + B_j P_{CP_j}^{CA}) + P_{CP}^{CAi}, \tag{2.4}$$

where $P_{TX_j} = \frac{P_{out_j}}{\eta}$ denotes the effective transmit power used by CC j. B_j and $P_{CP_j}^{CA}$ are represented as the bandwidth of CC j and the variable circuit power consumption, which scales linearly with both the number of active

CCs, and their bandwidth, B_j. However, P_{CP}^{CAi} is the static circuit power consumption of CA system.

2.4.2 Energy Consumption Models of DU and CU

Within the paper [45], two distinct power consumption models are delineated. The first centers on processing and is built upon the foundations of the EARTH model. This model seeks to establish a function representing the average CPU load, denoted as $\bar{l}$, for each Edge Processing Module (EPM) in a time slot t over a duration T. The modeling of this equation is presented as follows:

$$E_{epm,j}^{p}(t) = (\mathbb{I}_{(\bar{l}_{epm,j}^{t}>0)} P_{epm} + P_{epm}^{'} * \frac{\bar{l}_{epm,j}^{t}}{C_{epm}})T, \tag{2.5}$$

where P_{epm} is the power consumption of EPM where DUs are implemented, which represents the fixed costs of running the server, such as cooling, power amplification, and network switches. $P_{epm}^{'}$ is dynamic or load-dependent power consumption increases linearly with the EPM's load(i.e., average CPU load). $\mathbb{I}_{(\bar{l}_{epm,j}^{t}>0)}$ is an indicator variable that takes the value 1 when EPM j is busy and 0 when it is idle.

Similarly, the energy consumption of a Central Processing Module (CPM) k where CUs are implemented is given as:

$$E_{cpm,k}^{p}(t) = (\mathbb{I}_{(\bar{l}_{cpm,j}^{k}>0)} P_{cpm} + P_{cpm}^{'} * \frac{\bar{l}_{cpm,k}^{t}}{C_{cpm}})T, \tag{2.6}$$

where P_{cpm} and $P_{cpm}^{'}$ are static and dynamic power consumption respectively.

While the EARTH model offers insights into power consumption, it is not directly applicable for estimating the energy usage of the DU/CU. This

limitation arises primarily for two reasons[46, 47]. Firstly, considering that multiple DU/CUs operate within a cloud infrastructure, the energy footprint of an individual DU/CU is expected to be reduced. Secondly, due to the dynamic resource allocation inherent in O-RAN systems, applications residing on the DU/CU aren't continuously active.

Consequently, in the context of the O-RAN framework where the DU/CU essentially constitutes a segment of the virtualized BS, the primary determinant of its power consumption is attributed to the utilization of CPU cores [47]. Thus, the power consumption model of a DU/CU for a given time slot t can be articulated as follows:

$$P^t_{DU/CU} = N_c(P_{DU/CU,min} + \Delta_{P_{DU/CU}} \delta_c s^\beta), \tag{2.7}$$

where $\Delta_{P_{DU/CU}}$ is denoted as follow:

$$\Delta_{P_{DU/CU}} = \left(P_{DU/CU,max} - P_{DU/CU,min}\right) / s_0^\beta \tag{2.8}$$

In this model, the power consumption of DU is linear with the number of active CPU cores and CPU load as well. Where, N_c represents the number of active CPU cores in DU, $P_{m,min}$ and $P_{m,max}$ are denoted as the minimum and maximum power consumption of each CPU core.δ_c is the percentage of CPU load on active cores, s is the CPU speed and s_0 is the reference CPU speed. β is the exponential coefficient of CPU speed. Δ_{P_m} indicates the slope of the load-dependent power consumption of DU.

The percentage of CPU load on active cores, denoted as δ_c, can be ascer-

tained through the subsequent function:[47]:

$$\delta_c = \frac{Q(r)}{N_c s} = \frac{c_0 + kr}{N_c s} \tag{2.9}$$

Where $Q(r)$ is the actual instructions per unit time and $N_c s$ represents the maximum instructions available per unit time. c_0 is constant coefficient of instruction speed. k is rate varying coefficient of instruction. r is transmission rate. Base on eq (2.7) and eq (2.9), the power consumption model can be modeled as:

$$P^t_{DU/CU} = N_c P_{DU/CU,min} + \Delta_{P_{DU/CU}} c_0 s^{\beta-1} + \Delta_{P_{DU/CU}} krs^{\beta-1} \tag{2.10}$$

In [48], the authors proposed an energy consumption model of activating servers and instantiating O-RAN applications, representing as follows:

$$E_s(x_s, y_s) = x_s * E^{base}_s + \sum_{a\in\mathcal{A}} \sum_{r\in\mathcal{R}} y_{r,a,s} e_{a,s}, \tag{2.11}$$

where x_s indicates the server activation profile and E^{base}_s represents the fixed energy consumption when the server s is active. $y_s = (y_{r,a,s})_{(r,a)\in\mathcal{R}\times\mathcal{A}}$ which is an allocation variable indicates the load of server s, that is how many instances of app a with request r have been deployed on that server. $e_{a,s}$ shows the energy consumption of an application a which scales linearly with the server load.

Energy efficiency studies based on O-RAN are still in their nascent stages. Currently, only a few articles have considered and analyzed the Energy efficiency of O-RAN. Current analyses of O-RAN energy consumption models primarily rely on models developed for C-RAN and v-RAN. Although these network structures share similarities with O-RAN, they do not fully meet all

the requirements of O-RAN. Therefore, further research is needed to delve into energy efficiency optimisation that aligns with O-RAN's unique characteristics, aiming to directly reduce the overall energy consumption of O-RAN. For example, developing a distinct power consumption model unique to O-RAN, which includes the energy used during transmission and operation by various hardware and software components, rather than merely adopting or slightly modifying the existing EARTH model.

2.5 BS Sleep Mode Control Techniques

Among the many approaches proposed for improving network energy efficiency, BS sleep mode control remains one of the most direct and effective techniques. By dynamically switching lightly loaded nodes into low-power states, sleep control can significantly reduce energy consumption while maintaining acceptable QoS. However, the design of such mechanisms becomes increasingly challenging in dense and dynamic network environments. This section reviews the main categories of sleep mode control techniques, including traditional optimisation-based methods, centralized learning-based approaches, and decentralized learning-based frameworks, thereby positioning the specific research focus of this thesis.

2.5.1 Traditional Optimisation Methods

Wu *et al.* [4] proposed two traffic-aware sleep strategies, threshold-based and vacation-based, that jointly optimize BS activation and transmit power, revealing a fundamental tradeoff between energy savings and user delay. In

heterogeneous networks, energy savings can be achieved by selectively deactivating underutilized BSs. Zhou *et al.* [5] formulated a joint user association and BS operation framework to optimize long-term network utility under energy constraints. Their results show that activating only a subset of Small Base Stations (SBSs) based on traffic demand can significantly reduce energy use without degrading service quality. To incorporate renewable energy sources, Gong *et al.* [49] considered a two-stage optimisation scheme for BS sleeping and subcarrier allocation, aiming to minimize non-renewable power consumption. Their approach dynamically adjusts BS states based on both harvested energy availability and user traffic, striking a balance between energy cost and blocking probability. Recent architectural advances enable finer-grained sleep control. Sun *et al.* [50] investigated BS sleeping in uplink–downlink fully decoupled RANs, jointly optimizing user association, power allocation, and BS activity. Similarly, Wu *et al.* [51] modeled BS sleep behavior using a Semi-Markov Decision Process (SMDP), enabling dynamic sleep scheduling in macro-femto networks based on user arrival patterns and energy efficiency.

Although such optimisation-based approaches can be effective, their scalability degrades with growing network size and multi-objective constraints, leading to high computational complexity and long solution times. For example, [52] proposed heuristic xApps for RC ON–OFF control under static user deployment, where UEs were assumed to be non-mobile and decisions were made based on instantaneous load and RSS thresholds. While such approaches demonstrate energy savings in snapshot-like scenarios, the decision rules are manually designed and rely on fixed thresholds. Subsequently, [53]

introduced a DQN-based xApp under a similar static network setting and showed that learning-based policies outperform the heuristic model, even in stationary environments. This indicates that the RC activation problem is inherently combinatorial and difficult to fully capture through fixed rule-based logic.

2.5.2 Centralized Learning-based Optimisation Methods

The sleep/active mode decision problem is NP-hard due to the combinatorial on/off choices and temporal coupling, which makes exact optimisation intractable at scale. To address this challenge while satisfying QoS constraints, ML-based strategies have been widely studied, particularly in Ultra Dense Network (UDN) scenarios where spatial redundancy enables flexible sleeping. Paper [54] employs an LSTM-based predictor to capture temporal patterns in traffic and channel variations, enabling proactive on/off switching. In [9], a Deep Q-Learning Network (DQN) framework is enhanced with an action-selection network that filters out invalid actions and mitigates ineffective exploration. The work in [55] introduces a modular DRL pipeline that integrates a Deep Deterministic Policy Gradient (DDPG) policy with a supervised cost estimator and traffic predictor to account for switching costs and QoS degradation. Spatio–temporal correlations are further leveraged in [56], where convolutional-LSTM forecasting is coupled with an actor–critic controller to improve robustness under fluctuating traffic. A different direction is explored in [57], which formulates joint sleep control and renewable-energy sharing as a single Markov Decision Process (MDP), solved through a multi-

discrete Proximal Policy Optimization (PPO) that factorizes the exponential action space into tractable subspaces. Paper [58] presents a PPO-based approach that simultaneously considers sleep scheduling, cell zooming, user association, and Reconfigurable Intelligent Surface (RIS) configuration, thus addressing multi-cell coordination under mixed discrete/continuous actions. Risk-aware mechanisms have also been investigated. In [59], a digital twin is integrated with a DQN controller to assess delay risk before executing sleep decisions, triggering re-training or feature suppression under anomalous traffic conditions. From a multi-agent perspective, [60] treats each BS as an agent and applies a multi-agent PPO with a state-similarity heuristic, jointly optimizing BS sleeping and MIMO antenna operations while reducing oscillatory behavior. Recent works have explored heuristic and learning-based RC switching strategies within the O-RAN framework.

2.5.3 Decentralized Learning-based Optimisation Methods

However, centralized ML approaches typically collect training data at a single server, which raises scalability concerns in large, distributed systems: as RUs/BSs and UEs increase, the communication burden grows rapidly and data heterogeneity across locations hinders convergence and generalization [61]. Federated Learning (FL) addresses these issues by training locally and aggregating model updates instead of raw data, reducing backhaul load and improving locality adaptation [62]. Nevertheless, Federated Reinforcement Learning (FRL) is sensitive to client heterogeneity; inconsistent local policies can lead to conflicting updates and a less effective global policy [63].

Applying federated DRL to sleep control has shown promise. For example, [64] lets each SBS train a local Double Deep Q-Learning Network (DDQN) and periodically aggregates at a macro BS. It treats each SBS as an isolated agent, resembling a loosely coupled multi-agent system; and it aggregates at the macro-site level, which is agnostic to the functional split and control timescales defined in O-RAN. In contrast to prior cellular-centric FL/FRL designs, O-RAN introduces a native control hierarchy (Non-RT RIC for long-timescale policy/orchestration and Near-RT RIC for near-real-time control).

2.6 Summary

In summary, this chapter has reviewed the main technical foundations relevant to this thesis. It first traced the evolution of radio access network architectures from D-RAN to C-RAN and v-RAN, showing how the need for greater flexibility, programmability, and resource efficiency ultimately led to the emergence of O-RAN. It then examined the key innovations of O-RAN, including disaggregation, virtualization, open interfaces, and the RIC-based control framework, which together provide the architectural basis for intelligent and software-driven network management. Building on this foundation, the chapter reviewed how AI and ML can be integrated into O-RAN for network optimisation, and discussed existing models for characterising the energy consumption of O-RAN components, particularly the RU, DU, and CU. Finally, the chapter surveyed existing BS sleep mode control techniques, ranging from traditional optimisation methods to centralized and decentralized learning-based approaches.

Overall, the reviewed literature confirms that O-RAN creates a promising platform for intelligent energy-efficient control, but also reveals several important limitations in existing research. In particular, many current methods either rely on simplified control logic, suffer from scalability challenges in dense deployments, or do not explicitly incorporate sustainability objectives such as carbon-aware operation. These limitations directly motivate the work presented in the subsequent chapters of this thesis, which address practical xApp-based control, DRL-based RU sleep optimisation, scalable federated learning, and carbon-aware intelligent O-RAN operation.

3 Energy-Efficient xApp-Based Control in O-RAN

3.1 Introduction

In contrast to conventional optimization frameworks, O-RAN enables the deployment of intelligent control logic directly within the RAN through xApps operating on the Near-RT RIC. This innovation provides a practical pathway for implementing real-time energy-efficient network control mechanisms. In this context, this chapter focuses on the design and implementation of practical xApp-based mechanisms for sleep mode control in O-RAN systems. Specifically, we consider the control of RCs within O-RUs, which are responsible for radio-frequency operations and represent a significant portion of the overall energy consumption. By leveraging real-time network measurements provided through the O-RAN interfaces, the proposed xApps dynamically adjust the operational states of RCs according to traffic demand, signal quality, and resource utilization conditions.

Two lightweight xApps are developed in this chapter. The first xApp targets idle RCs and transitions them into sleep mode when no active users are associated, thereby eliminating unnecessary energy consumption. The second xApp extends this approach by incorporating load-aware decision-

making, where lightly loaded RCs are selectively deactivated through user reassignment, subject to QoS and capacity constraints. Compared with optimisation-based or learning-based approaches, these methods are computationally efficient and can be directly deployed in operational environments without requiring extensive training or model calibration. To validate the practicality of the proposed approach, the xApps are implemented and evaluated using a commercial-grade RAN emulator – VIAVI AI-RSG [65], which emulates realistic O-RAN network conditions and interfaces. The experimental results demonstrate that significant energy savings can be achieved, particularly under low and moderate traffic loads, confirming the effectiveness of rule-based xApp control in real-world scenarios.

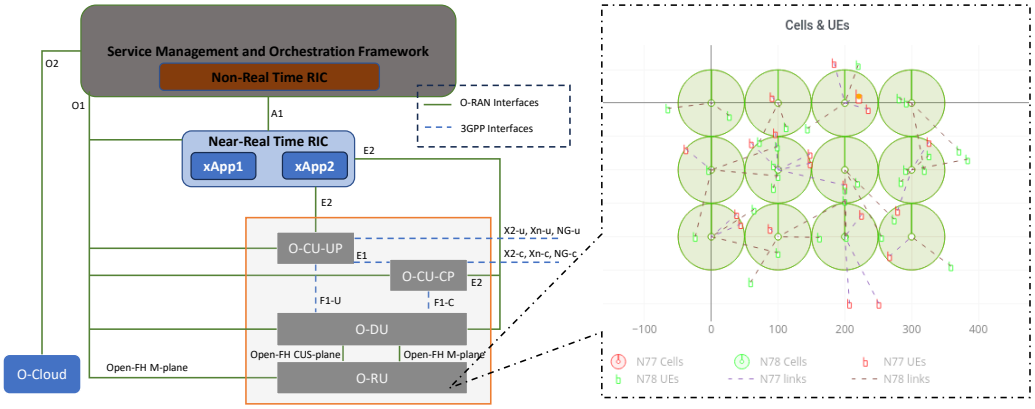


Figure 3.1: shows O-RAN Architecture, with components and interfaces from O-RAN and 3GPP. O-RAN interfaces are drawn as solid lines, 3GPP ones as dashed lines. The layout of the O-RU and the distribution of users is shown in the picture in the right section. Each O-RU consists of two RCs operating on distinct frequencies, n77 and n78, represented by red and green circles, respectively. Due to the identical coverage areas of both RCs, only the color of one RC is discernible in the figure. The color of the UEs within the figure matches the RC to which they are connected, indicating their network association based on the respective RC's frequency. The proposed two xApps are anchored in Near-RT RIC

3.2 System model

The configuration of the system implemented in this study is delineated. To craft a simulation environment that mirrors real-world conditions closely, the 'TeraVM' RIC tester is employed. Additionally, the Urban Microcell path loss (UMa) channel model is utilized to enhance the realism of the simulations [65]. This research incorporates a single O-CU and 12 O-RUs, each of which is serviced by a corresponding O-DU. The O-RUs are positioned at fixed locations, maintaining a uniform inter-site distance of 100 meters. The installation height for each O-RU is standardized at 10 meters, and each O-RU encompasses two RCs operating on distinct frequencies—specifically, n77 and n78 as illustrated in Fig. 3.1. Both RCs are equipped with isotropic antennas, and the transmission power for each RC is set to 30 dBm. During the simulation, all the UEs are static and randomly distributed in the considered area. For the sake of subsequent better expression, we assume that there are a set of RCs $\mathbf{M}$ and a set of UEs $\mathbf{K}$. Let $k \in \mathbf{K}$ be the index of k-th UE and $m \in \mathbf{M}$ be the index of m-th RC. The power consumption model for RCs are delineated following the framework presented by [66]. The model encapsulates the various components of power expenditure as follows:

$$P_{RC} = P_{FIX}^{mode} + P_{data} + P_{TC}, \tag{3.1}$$

where, P_{RC} represents the overall power consumption of all RCs in the network. P_{TC} denotes for the power consumed by the transceiver chains. The term P_{FIX}^{mode} represents the fixed power consumption of RCs, which varies depending on whether an RC is in its active state or sleep mode. This can be

expressed as:

$$P_{FIX}^{mode} = \sum_{m=1}^{M} (\alpha_m * P_{Active} + (1 - \alpha_m) * P_{Sleep}), \tag{3.2}$$

in which α_m is a binary indicator denoting the operational state of RC m (1 for active and 0 for sleep). The P_{Active} and P_{Sleep} respectively signify the fixed power consumption rates of an RC in its active and sleep states. Furthermore, accounts for the power consumption associated with the transmission power of RC m, which scales with the usage of RC resource blocks as defined by:

$$P_{data} = \sum_{m=1}^{M} \alpha_m * \frac{P_{tx}}{\eta} * PRB_{usage}^{m}, \tag{3.3}$$

where the P_{tx} is the transmission power of RC and η is represented as the PA efficiency. The term PRB_{usage}^{m} reflects the resource block usage on RC m, illustrating that an increase in the number of UEs served by an RC directly influences its power consumption.

3.3 Problem Formulation

The objective function of this study is to minimize the total power consumption of the network. To achieve this, we have developed two xApps that adjusts the activity of RCs based on current network conditions. The xApps aim to find the most energy-efficient configuration of an RCs without compromising network performance and quality. This optimization is subject to three key constraints. The first is Reference Signals Received Power (RSRP), which represents the power of the reference signals spread over the entire

bandwidth and is measured in decibels milliwatt (dBm). RSRP is a vital indicator of signal strength received by a UE from an RC and plays a significant role in determining the quality of the connection between the UE and the network. By ensuring that each UE maintains a minimum UE level, we aim to guarantee robust and reliable connectivity across the network.

In addition, data rate as the second constrain ensures that the data rate meets the operational needs of the UEs. The last constrain is managing resource block usage on each RC efficiently. This resource block management prevents any RC from becoming overwhelmed with traffic, ensuring a balanced distribution of network resources. Consequently, the resource block utilization L_m of RC m is defined as the average portion of bandwidth occupied:

$$L_m = \sum_{k \in \mathbf{K}_m} B_k / B_m, \tag{3.4}$$

where B_m denotes the maximum resource block of RC m and B_k represents the resource blocks currently occupied by the UE k. Hence, the problem of minimizing the network's power consumption is formulated as:

$$\begin{aligned}
\mathcal{P}_1: \quad \min \quad & P_{RC} \\
\text{s.t.} \quad & \sum_{m=1}^{M} \beta_{k,m} \gamma_{k,m} \geq \gamma_{min}, \quad \forall k \in \mathbf{K}, \\
& \sum_{m=1}^{M} \beta_{k,m} R_{k,m} \geq R_{min}, \quad \forall k \in \mathbf{K}, \\
& \sum_{m=1}^{M} \beta_{k,m} \leq 1 \quad \forall k \in \mathbf{K}, \\
& L_m \leq 1, \quad \forall m \in \mathbf{M} \\
& \beta_{k,m} \in \{0,1\}, \quad \forall k \in \mathbf{K}, \forall m \in \mathbf{M},
\end{aligned} \tag{3.5}$$

where $\beta_{k,m}$ is the binary variable to represent the association between the UE k and RC m. $\beta_{k,m} = 1$ means UE k and RC m is associated. γ_{min} is the minimum RSRP threshold necessary for UE k and $\gamma_{k,m}$ is the RSRP of UE k from RC m. $R_{k,m}$ is the data rate of UE k when associate with RC m and R_{min} is the minimum required data rate by a UE. A UE is considered to be in an outage if its data rate is below this specified threshold. Additionally, to prevent any RC from being overloaded, L_m must not exceed 1.

3.4 RC Switching Algorithm

Problem $\mathcal{P}_1$ defines the objective of minimizing the total RC power consumption subject to signal quality, data rate, association, and load constraints. In this chapter, $\mathcal{P}_1$ is not solved through an exhaustive or globally optimal optimisation method. Instead, xApp 1 and xAapp 2 provide practical heuristic procedures to obtain feasible lower-power RC configurations. The main idea is to reduce the number of active RCs, since switching an RC from active mode to sleep mode directly reduces the fixed power term P_{FIX}^{mode} and removes the corresponding load-dependent power consumption in P_{data}.

xApp 1 considers the most conservative case by switching an RC into sleep mode only when it has no associated UEs. This reduces P_{RC} without changing user association or affecting the constraints in $\mathcal{P}_1$. xApp 2 further reduces power consumption by identifying lightly loaded RCs and attempting to reassign their associated UEs to other active RCs. A candidate RC is switched into sleep mode only if all its served UEs can be reassigned while satisfying the minimum RSRP constraint, the minimum data rate constraint, and the load constraint of the target RCs. Therefore, the proposed algorithms

address $\mathcal{P}_1$ by greedily searching for feasible RC deactivation decisions that reduce the objective function while maintaining the required network performance constraints.

3.4.1 Proposed xApp1

Given that UEs are randomly distributed within a designated area, this results in not all RCs being associated with UEs. Especially when the number of UEs is relatively small, there will be some RCs in idle state (i.e., not serving UEs). Our designed first xApp is to switch those RCs in to sleep mode, which illustrate on Algorithm 1. First of all it iterates through all the RCs in the RC list and then utilizes the KPM of RRC.ConnMean to assess the number of UEs serviced by each RC [67]. If an RC is found to have no UEs to serve. The xApp to transition that RC into sleep mode.

Algorithm 1: xApp1 algorithm

```
for each RC in M do
    Retrieve the list of UEs attached to the current RC
    if the list of attached UEs is empty then
        Put the current RC into sleep mode
```

3.4.2 Proposed xApp2

The Algorithm 2 is proposed based on the initial intervention by the first xApp, certain RCs are set to sleep mode to enhance network efficiency. This action, however, leads to a new scenario where some of the still-active RCs are found to be serving a very small number of UEs, indicating a low-load operation. These RCs, due to their minimal usage, become prime candidates

for potentially entering sleep mode to conserve power. In the proposed algorithm, when both of resource block usage and throughput below the certain threshold can be regarded as low-load RCs.

The possibility of transitioning these low-load RCs into sleep mode hinges on successfully reallocating the UEs they serve to other RCs. This reallocation must meet two critical conditions: firstly, the RSRP for the UEs must remain above a certain threshold to ensure no compromise on service quality; secondly, the load on any new RC tasked with serving additional UEs should not exceed a specific limit, for instance, 50%, to prevent overburdening. When all the UEs in the RC have been successfully reassigned out, the RC can enter the sleep state. The reassigned UE needs to update its RSS based on the new association. But when the UE cannot fulfil both hard constrains. The RC must remain active even if it is in a low load state.

The proposed algorithm seeks to further diminish the network's total power usage. This ensures that the network capacity is utilised appropriately without unnecessarily keeping under-loaded RCs active. This xApp presents a balanced way to optimize network performance while minimizing power expenditure.

The two proposed xApps in this chapter can be regarded as heuristic or rule-based control methods. Instead of learning a policy from data, the control decisions are made according to predefined operational rules derived from network engineering considerations. In xApp1, the rule is straightforward: an RC is switched into sleep mode only when it has no associated UEs. This represents a conservative heuristic because it avoids any reassignment procedure and therefore has minimal risk of degrading user service. In

xApp2, a more aggressive heuristic is used. An active RC is considered as a candidate for sleep mode when its resource block utilisation and carried traffic are both below predefined low-load thresholds. The UEs served by this RC are then reassigned only if the target RC can satisfy the minimum RSRP requirement, the minimum data rate requirement, and the maximum load constraint. In this study, the maximum load threshold of the target RC is set to 50%, so that reassignment does not overload the target RC and sufficient residual capacity is preserved for traffic variation.

The choice of these thresholds reflects a trade-off between energy saving and service protection. A more relaxed threshold allows more RCs to be switched off and can therefore improve energy saving, but it may increase the risk of congestion, poor signal quality, or user outage after reassignment. Conversely, a stricter threshold is more conservative and better protects QoS, but it may leave lightly loaded RCs active and reduce the achievable power saving. Therefore, the thresholds in the heuristic xApps are selected to prioritise QoS satisfaction while still enabling energy reduction under low and moderate traffic loads.

Compared with AI-based approaches, heuristic methods have several practical advantages. They are simple to implement, computationally lightweight, interpretable, and do not require training data or offline model calibration. These properties make them suitable for initial deployment in Near-RT RIC xApps, especially when the control logic must be transparent and predictable. However, heuristic methods are also limited by their fixed rules and manually selected thresholds. They may not adapt well to highly dynamic traffic, mobility, or heterogeneous network conditions. AI-based methods, such as rein-

forcement learning, can learn adaptive control policies from interaction with the environment and may achieve better long-term performance in complex scenarios. Nevertheless, AI-based methods require training, careful reward design, and additional computational resources, and their decisions may be less interpretable. For this reason, the heuristic methods in this chapter provide a practical baseline and deployment-oriented starting point, while the following chapters investigate AI-based approaches for more adaptive and scalable RU sleep control.

Algorithm 2: xApp2 algorithm

```
Initialize Algorithm 1
for each active RC m do
    if the resource block usage and throughput of RC m less than
     threshold then
        Select those RCs
        for each UE k served by selected RC m do
            for each RC n(n ∈ M, n ≠ m) in the network do
                if the RSRP of UE k is satisfied threshold γ_min and
                 load of RC L_n can accept extra UE then
                    Reassign the UE to RC n and update the UE's
                     RSRP based on the new association.
                else
                    RC m can't be switched into sleep mode
        if all the UE k served by RC m are re-associated then
            Switch RC m into sleep mode
```

3.5 Results

We consider a network consisting of 12 O-RUs and 24 RCs in the area of 500 $\times$ 500 m^2, where the inter-site-distance is 100m. We randomly distributed

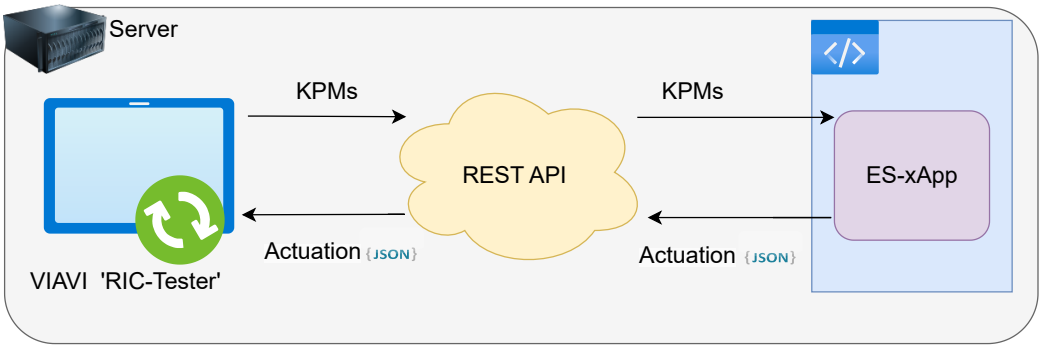


Figure 3.2: shows the flowgraph that how the python code based xApp interact with RIC-tester.

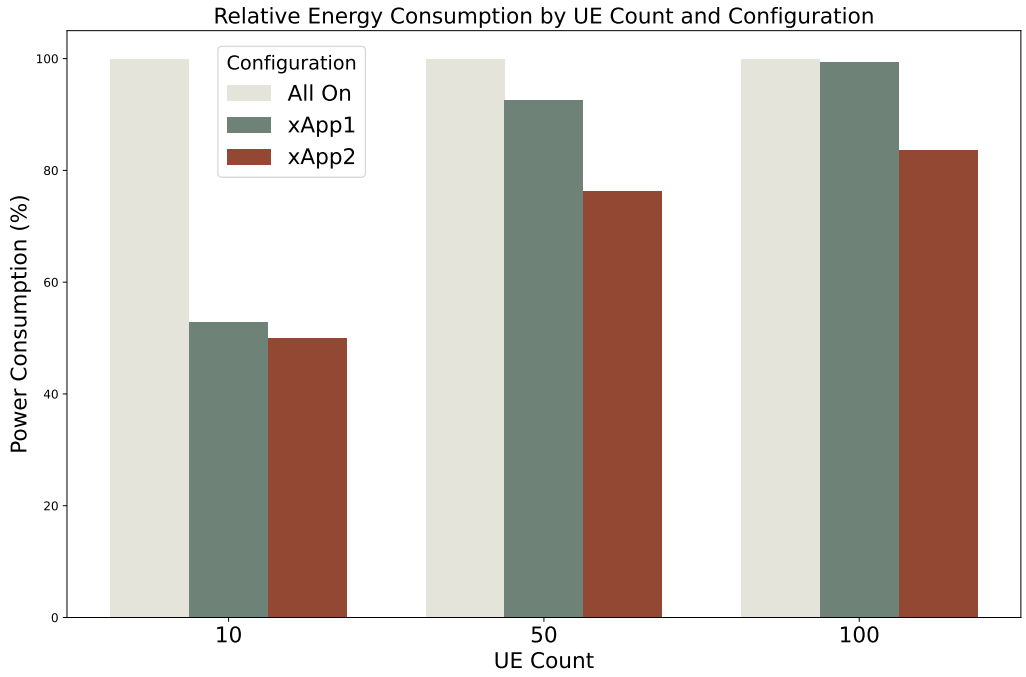


Figure 3.3: shows the power consumption in percentage among 10, 50 and 100 UEs in three different scenarios. All on is always in 100% power consumption. xApp1 and xApp2 show a relative percentage of the All on scenario.

10, 50 and 100 UEs. The channel model is chosen to follow the UMa model [68].

The entire procedure is systematically depicted in Fig. 3.2. Initially, an

Table 3.1: Simulation Parameters

Parameter	Value
Carrier frequency $n77/n78$	3.5/3.7GH
Channel bandwidth, B_m	100MHz
Fixed power in active mode, P_{Active}	20W
Fixed power in sleep mode, P_{Sleep}	5W
Power Amplifier efficiency (PA), η	1.67%
Power per RF chain, P_{TC}	1W
UE required data rate, R_{min}	10 Mbps
The number of O-RUs, $\mathbf{N}$	12
The number of RCs, $\mathbf{M}$	24

xApp initiates the process by dispatching a request for network data or KPMs to the RIC-tester through the REST API. Subsequently, the RIC-tester processes this request and furnishes the requisite feedback to the xApp. Upon receiving the feedback, the xApp evaluates the appropriate action based on its embedded algorithm and communicates the decision back to the RIC-tester using the 'REST' API. Finally, the RIC-tester implements the specified configuration in accordance with the action determined by the xApp.

Our initial exploration focuses on assessing the average (over 10 trials) power consumption in percentage across varying distributions of UE utilizing our proposed algorithms. Illustrated in Fig. 3.3, with a scenario of merely 10 UEs, both the xApp1 and the xApp2 demonstrate substantial power savings, reaching up to 50%. In this context, the performance differential between the xApp1 and xApp2 is less than 5% because with a small number of UEs, xApp1 has already switched most of the RCs into sleep mode, and xApp2 can no longer find RCs that fulfil the conditions. However, as the UE count escalates, a divergence in performance becomes apparent. Specifically, when the UE tally extends to 100, the xApp1's power consumption parallels that

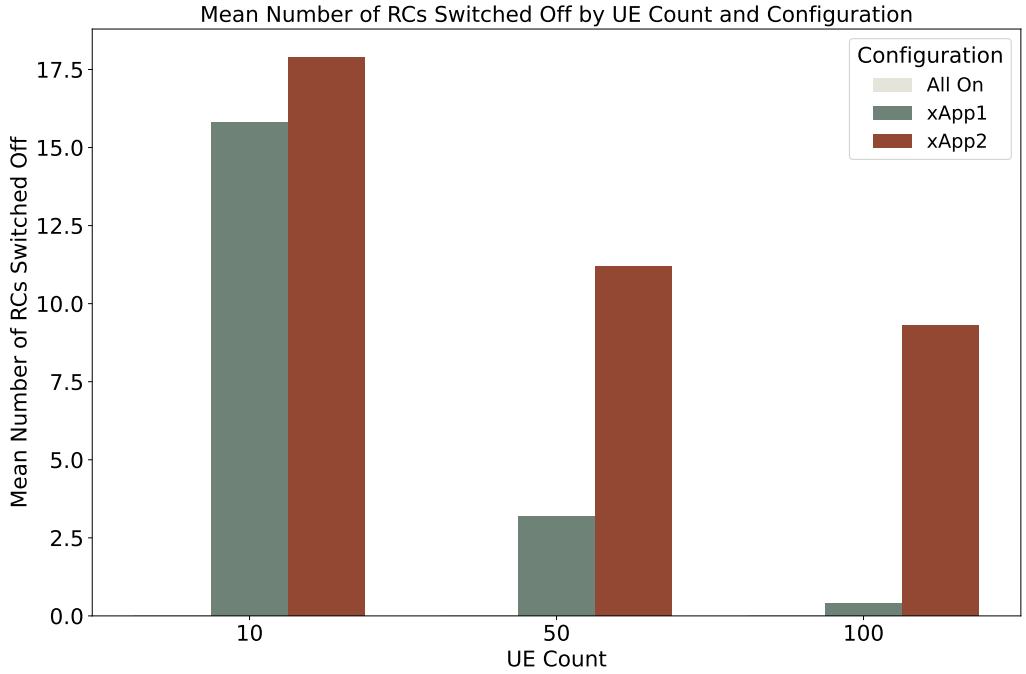


Figure 3.4: shows the average (over 10 trials) number of RCs are switched into sleep mode between 10, 50 and 100 UEs in three different scenarios

observed when all RCs are active, contrasting sharply with xApp2, which sustains approximately 15% in power savings. On top of that, xApp2 still outperforms xApp1 by almost 20%. This analysis underscores the efficiency of xApp2 in managing power consumption, particularly as network demand intensifies.

In Fig. 3.4 we examine the number of RCs are switched into sleep mode. It becomes evident that under conditions of high UE density, the xApp1 exhibits a limited propensity to transition RCs into sleep mode. This observation is corresponding to the data presented in Fig. 3.3. Conversely, the xApp2, leveraging our proposed algorithm, demonstrates the capability to place approximately 10 RCs into sleep mode, even with increased UE density (50 and 100). A comparative analysis of Fig. 3.3 and Fig. 3.4 reveals that,

for xApp2, the variation in the number of RCs entering sleep mode between scenarios with 50 and 100 UEs is marginal, at around 2 RCs. Nonetheless, this small difference leads to a more noticeable difference in power savings. This phenomenon can be attributed to the escalating demands placed on RCs as the number of UEs increased. Not only are RCs required to accommodate a larger volume of UEs, but they must also manage UEs located at greater distances. This scenario necessitates increased resource block usage utilization and, consequently, leads to heightened power consumption.

3.6 Summary

In this chapter, we introduced two xApps specifically designed for switching RCs into sleep mode while satisfying UE required data rate and resource block usage of RCs. This work also tested within the "TeraVM" RIC-tester, a commercial-grade simulation environment. Our numerical results substantiate the effectiveness of the proposed xApps in achieving considerable power savings across varying network demands. Moving forward, our focus will include considering user mobility, thereby treating traffic variations in O-RUs as an ongoing dynamic process. Machine learning techniques, such as deep reinforcement learning, are expected to offer superior performance in navigating the complexities and constraints of network environments.

4 Deep Reinforcement Learning for RU Sleep Control in O-RAN

4.1 Introduction

Building upon the practical xApp-based control strategies presented in the previous chapter, this chapter investigates the use of DRL to enable more adaptive and scalable energy-efficient operation in O-RAN systems. While rule-based approaches can achieve noticeable energy savings with low complexity, their reliance on predefined thresholds limits their effectiveness in dynamic and complex network environments with varying traffic, user mobility, and channel conditions.

To address these limitations, the RU sleep control problem is formulated as a sequential decision-making task and solved using DRL. In this framework, the network state captures key information such as user distribution, traffic load, and RU operational modes, while control actions determine the activation or deactivation of RUs. However, applying DRL to this problem introduces significant challenges, particularly in terms of action space scalability. Conventional discrete-action methods such as DQN suffer from exponential growth in action space as the number of RUs increases, making them impractical for dense O-RAN deployments. To overcome this limitation, this

chapter adopts the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, which enables efficient control in continuous action spaces. By mapping RU activation decisions into a continuous domain and applying discretisation, TD3 avoids combinatorial action explosion while maintaining effective control performance.

Based on this framework, a TD3-based xApp is developed for real-time RU sleep mode control, incorporating realistic system dynamics such as user mobility, resource allocation, and power consumption. Two DQN-based approaches are also implemented as baselines for comparison. The results demonstrate that the proposed TD3 method achieves superior energy efficiency, faster convergence, and better scalability compared to DQN-based methods, particularly in larger and more complex network scenarios.

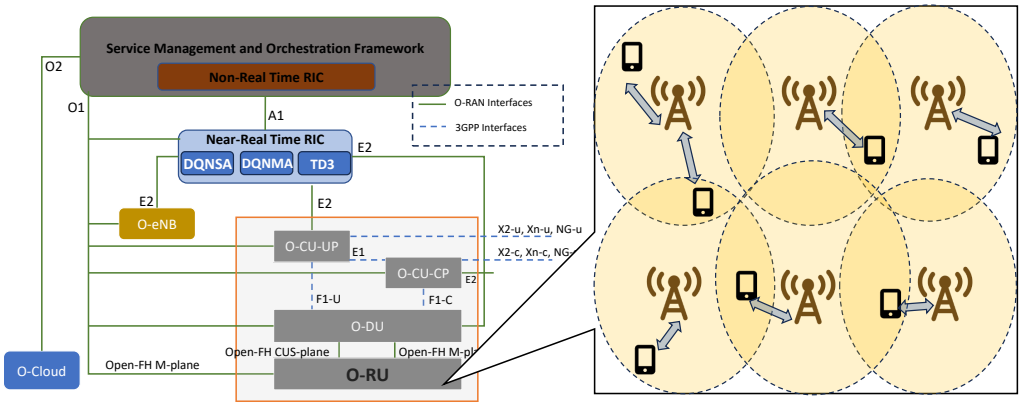


Figure 4.1: shows O-RAN Architecture, with components and interfaces from O-RAN and 3GPP. O-RAN interfaces are drawn as solid lines, 3GPP ones as dashed lines. The layout of the O-RU is shown in the picture in the right section.

4.2 Foundations of Deep Reinforcement Learning for RU Sleep Control

This chapter employs DRL to address the RU sleep mode control problem in dynamic O-RAN environments. In contrast to rule-based methods, DRL enables an agent to learn control policies directly from interactions with the environment, without requiring an explicit analytical solution to the underlying optimization problem. Since RU sleep control is inherently sequential, state-dependent, and affected by uncertain user mobility and traffic variations, it can be naturally formulated within the reinforcement learning framework. This section introduces the theoretical foundations of DRL that underpin the proposed method, including the MDP, value-based learning, the limitations of DQN in large-scale RU control, and the motivation for adopting the TD3 algorithm.

4.2.1 Markov Decision Process Formulation

A reinforcement learning problem is commonly modelled as a MDP[69], defined by the tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma), \tag{4.1}$$

where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s'|s,a)$ is the state transition probability, $r(s,a)$ is the immediate reward function, and $\gamma \in [0,1)$ is the discount factor.

At each time step t, the agent observes the current state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ according to a policy $\pi(a|s)$, receives an immediate reward r_t, and transitions to the next state s_{t+1}. The objective of the agent is to

learn a policy that maximizes the expected discounted return:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}. \tag{4.2}$$

Accordingly, the optimal policy can be defined as

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}[G_t]. \tag{4.3}$$

In the context of RU sleep control, the MDP formulation is particularly suitable because the network controller must repeatedly decide which RUs should remain active or be switched to sleep mode according to the current network condition. These decisions influence not only the instantaneous energy consumption, but also future user association, load distribution, and QoS satisfaction. Therefore, the problem is not a one-shot optimisation problem, but a sequential control problem with long-term consequences.

4.2.2 Value Functions and Bellman Equations

The foundation of reinforcement learning lies in the use of value functions to estimate the long-term benefit of states and actions. For a policy π, the state-value function is defined as

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s\right], \tag{4.4}$$

which represents the expected return obtained when starting from state s and thereafter following policy π.

Similarly, the action-value function, or Q-function, is defined as

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \,\middle|\, s_t = s, a_t = a\right]. \tag{4.5}$$

The optimal state-value and action-value functions satisfy the Bellman optimality equations:

$$V^{*}(s) = \max_{a\in\mathcal{A}}\left[r(s,a) + \gamma\sum_{s'}\mathcal{P}(s'|s,a)V^{*}(s')\right], \tag{4.6}$$

and

$$Q^{*}(s,a) = r(s,a) + \gamma\sum_{s'}\mathcal{P}(s'|s,a)\max_{a'}Q^{*}(s',a'). \tag{4.7}$$

These equations provide the theoretical basis for learning optimal control policies. However, in practical wireless network problems, the state transition probability $\mathcal{P}(s'|s,a)$ is unknown and the state space is usually too large for tabular methods. This motivates the use of approximate function learning through neural networks.

4.2.3 From Q-Learning to Deep Q-Networks

Classical Q-learning [70] updates the action-value estimate iteratively according to

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_t + \gamma\max_{a'}Q(s_{t+1},a') - Q(s_t,a_t)\right], \tag{4.8}$$

where α is the learning rate. The term

$$\delta_t = r_t + \gamma\max_{a'}Q(s_{t+1},a') - Q(s_t,a_t) \tag{4.9}$$

is commonly referred to as the Temporal Defence (TD) error.

When the state space becomes large or continuous, tabular Q-learning is no longer practical. DQN address this problem by approximating the Q-function with a neural network $Q(s, a; \theta)$ parameterized by θ. The network is trained by minimizing the loss

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[\left(y^{\mathrm{DQN}} - Q(s, a; \theta) \right)^2 \right], \tag{4.10}$$

where $\mathcal{D}$ denotes the replay buffer and the target is

$$y^{\mathrm{DQN}} = r + \gamma \max_{a'} Q(s', a'; \theta^-). \tag{4.11}$$

Here, θ^- represents the parameters of the target network [71], which are periodically updated to stabilize training.

DQN is effective in many discrete-action problems. However, its applicability to RU sleep control is fundamentally constrained by the structure of the action space.

4.2.4 Limitations of DQN for RU Sleep Control

In the RU sleep control problem, each RU can take one of two operational modes: active or sleep. If the network consists of M RUs, then the number of possible joint activation patterns is $|\mathcal{A}| = 2^M$. This means that the action space grows exponentially with the number of RUs. For example, when $M = 12$, the controller must in principle evaluate $2^{12} = 4096$ possible actions at each time step. As M increases further, this becomes computationally prohibitive for DQN, because the maximisation over all actions in the target

$\max_{a'} Q(s', a'; \theta^-)$ must still be performed.

This issue is particularly problematic in dense O-RAN deployments, where a large number of RUs must be jointly controlled. Although simplified DQN variants may reduce complexity by updating only part of the action vector at each time step, such strategies typically sacrifice global coordination and may lead to suboptimal control decisions. Therefore, a more scalable reinforcement learning framework is required for large-scale RU sleep mode optimisation.

4.2.5 Policy-Based and Actor–Critic Reinforcement Learning

An alternative to value-based methods is to directly learn a policy. In policy-based reinforcement learning, the controller is represented by a parameterised policy $\pi_\theta(a|s)$, and the objective is to maximise the expected return:

$$J(\theta) = \mathbb{E}_{\pi_\theta}[G_t]. \tag{4.12}$$

The policy parameters can be updated using the policy gradient theorem:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)\, Q^\pi(s, a)\right]. \tag{4.13}$$

Pure policy-gradient methods can suffer from high variance. To improve learning efficiency, actor–critic methods introduce two separate function approximators: an *actor*, which learns the policy, and a *critic*, which estimates the value of the policy. In this framework, the critic provides feedback to guide the actor update, thereby combining the advantages of value-based and

policy-based learning.

For continuous control problems, deterministic policy gradient methods are often more efficient. Instead of learning a stochastic policy $\pi_\theta(a|s)$, the actor learns a deterministic mapping

$$a = \mu_\theta(s), \tag{4.14}$$

where μ_θ directly outputs the control action. The deterministic policy gradient is given by

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_a Q(s,a)|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s)\right]. \tag{4.15}$$

This formulation is especially attractive when the control dimension grows, since the actor can output a vector of actions without requiring explicit enumeration of all discrete combinations.

4.3 System Model

In this section, we discuss the system model of O-RAN environment as demonstrated at 4.1. We consider the downlink transmission where M RUs equipped with a single antenna serve K UEs. Each RU is served by a corresponding DU and DUs are connected with single CU. In addition, UEs move at a speed v, choosing from 0 to v_{max}. The mobility of the UEs is modelled using a bounded movement pattern within the considered service area. The speed of each UE is generated from a normal distribution within the predefined speed range and is treated as a discrete simulation value. Once

assigned, the speed of each UE remains unchanged during the simulation. At the beginning of the simulation, the UEs are randomly generated at the four corners of the considered area and then move towards the centre of the area. When a UE reaches the boundary of the service area, its moving direction is reversed, so that the UE continues to move within the considered area. Note that, in our study, the UEs move within the area and do not leave it. The sets of RUs and UEs are represented as $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$ respectively. We also introduce the α_m which is a binary variable to indicate the active/sleep of RU m. Here $\alpha_m = 1$ means the RU m is in active mode, otherwise $\alpha_m = 0$.

In our work, each RU is able to provide a total of Q_m Physical Resource Blocks (PRBs) to the UEs associate to the RU. Let U_m^t denote the total number of UEs associate with RU m at time slot t. On top of that, we denote $n_{m,k}^t$ is the number of PRBs allocated to the k^{th} UE in RU m. Therefore, the total number of PRBs usage N_m^t of RU m at time slot t can be computed as $N_m^t = \sum_{u=0}^{U_m^t} n_{m,k}^t$. Based on this, the RU load at time slot t is defined as:

$$l_m^t = \frac{N_m^t}{Q_m}. \tag{4.16}$$

In our work, we consider the fading channel model from the RU m to the UE k as $h_{m,k}$. Therefore the Signal-to-Noise Ratio (SNR) $SNR_{m,k}$ of the k^{th} UE associate to the RU m at time t is presented as :

$$SNR_{m,k} = \frac{P_{TX} h_{m,k} \frac{n_{m,k}}{Q_m}}{n_{m,k} N_0}, \tag{4.17}$$

where P_{TX} is the maximum transmission power of an RU and the transmis-

sion power allocated to the UE depends on the percentage of PRBs occupancy. The denominator corresponding to additive white Gaussian noise. The corresponding data rate of UE k is given by:

$$R_k^t = n_{m,k} B * \log_2(1 + SNR_{m,k}), \tag{4.18}$$

where B is the bandwidth of each PRB

4.3.1 Power Consumption Model and Problem Formulation

The power consumption of RU is divided into three parts. The fisrt part is the fixed power P_m^{Fix} of RUs consumed by the signal processing, cooling system and power supply, which only varies depending on the active/sleep mode of RU. Then the load dependent power comes from the PA P_m^{data}. The load in this paper is represented by the PRB usage as shown in (4.16), and transition power P_m^{trans} is generated when the mode of the base station changes, e.g. from sleep mode to active mode. Combining these three parts, the power consumption of the RU m at time slot t is:

$$P_m^t = P_m^{Fix,t} + P_m^{data,t} + P_m^{trans,t}. \tag{4.19}$$

First, fixed power of RU m consumed by the active or sleep mode is given by:

$$P_m^{Fix,t} = \alpha_m^t P_m^{active} + (1 - \alpha_m^t) P_m^{sleep}, \tag{4.20}$$

where P_m^{active} and P_m^{sleep} are represented the fixed power consumption of RU m in active mode and sleep mode, respectively. Then, the transmission power

of RU m which is scaled with load of RU m at time slot t computed as:

$$P_m^{data,t} = \alpha_m^t \frac{P_{TX}}{\eta} * l_m^t = \alpha_m^t \frac{P_{TX}}{\eta} * \frac{N_m^t}{Q_m}, \tag{4.21}$$

where $\eta \in [0, 1]$ is the power amplifier efficiency of the RU m. P_{TX} here is the constant value and only scale with the percentage of the load. Therefore, the more UEs are associated the more power consume. Last but not least, the transition power of RU m is given by:

$$P_m^{trans,t} = \max\left(\alpha_m^t - \alpha_m^{t-1}, 0\right) \cdot V_m^{trans} \tag{4.22}$$

where, α_m^{t-1} is the active/sleep mode of RU m in the last time slot. V_m^{trans} is the power consumed by switching on and off mode of the RU m. The transition power P_m^{trans} is only consumed when the RU is switched on.

Finally, the total power consumption of the entire network in time slot t is defined as:

$$\begin{aligned} P_{\text{tot}}^t &= \sum_{m=1}^{M} (P_m^{Fix,t} + P_m^{data,t} + P_m^{trans,t}) \\ &= \sum_{m=1}^{M} (\alpha_m^t P_m^{active} + (1 - \alpha_m^t) P_m^{sleep}) + \alpha_m^t \frac{P_{TX}}{\eta} * \frac{N_m^t}{Q_m} \\ &\quad + (\max\left(\alpha_m^t - \alpha_m^{t-1}, 0\right) \cdot V_m^{trans}) \end{aligned} \tag{4.23}$$

We assume all the RUs are in active at the initial stage when $t - 1 = 0$. Therefore, the power minimization problem can be formulated as follow:

$$\mathcal{P}_1 : \quad \min_{\{\alpha_m^t, n_{m,k}^t\}} \quad \sum_{t=1}^{T} P_{tot}^t$$
$$\text{s.t.} \quad R_k^t \geq R_{k,min}^t, \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T} \tag{4.24}$$
$$N_m^t \leq Q_m^t, \quad \forall m \in \mathcal{M}, \forall t \in \mathcal{T}$$
$$\alpha_m^t \in \{0, 1\}, \quad m \in \mathcal{M}, \forall t \in \mathcal{T},$$

where $R_{k,min}^t$ is the minimum data rate requirement of UE k at time interval t. The second constraint is that the allocated PRBs need to be within the total number of PRBs and $\mathcal{T} = \{1, 2, \ldots, T\}$.

4.4 Energy Efficiency DRL based Solution

The primary objective of this study is to optimize the sleep and active modes of BS in dynamic radio environments to minimize network-wide energy consumption without impacting the UEs QoS. In this section, we proposed a solution combined with TD3 algorithm, an advanced actor-critical RL framework to optimize the sleep mode strategy.

4.4.1 Markov Decision Process Problem

In this subsection, we discuss the state space, action space, and reward function of DRL model to consist the MDP problem.

State Space

state comprises essential information used for policy training. At each time step t, it contains the real data rate of the UEs , represented as:

$$\mathbf{R}_d^t = \left[R_{d,1}^t, R_{d,2}^t, \ldots, R_{d,K}^t\right]^T, \tag{4.25}$$

which captures the system's ability to meet user demands. The activation states of the radio units (RUs) from the previous time step are denoted as:

$$\boldsymbol{\alpha}^{t-1} = \left[\alpha_1^{t-1}, \alpha_2^{t-1}, \ldots, \alpha_M^{t-1}\right]^T. \tag{4.26}$$

The PRB utilization of the RUs is given by:

$$\mathbf{L}^t = \left[l_1^t, l_2^t, \ldots, l_M^t\right]^T, \tag{4.27}$$

quantifying the load on the RUs. Finally, the spatial positions of the UEs in the network are represented as:

$$\mathbf{U}^t = \left[(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_K^t, y_K^t)\right]^T, \tag{4.28}$$

characterized by their (x, y) coordinates. The UE coordinates determine path loss and association and thus affect data rates and RU loads. Including them in s_t enables the agent to capture the spatial distribution and mobility of UEs. This allows the learned policy to anticipate future handovers, traffic shifts between RUs, and coverage-critical regions, instead of reacting only to instantaneous loads.

In summary, the state can be expressed as:

$$s_t = \left[\mathbf{R}_d^t, \boldsymbol{\alpha}^{t-1}, \mathbf{L}^t, \mathbf{U}^t\right]^T. \tag{4.29}$$

All state parameters are normalized to facilitate a better interpretation by the agent.

Action Space

action $\alpha \in \mathcal{A}$ represents the binary operational states of all RUs, defined as:

$$\mathcal{A} = [\alpha_1, \alpha_2, \ldots, \alpha_M]. \tag{4.30}$$

Each element α_m corresponds to an RU, where $\alpha_m = 1$ denotes that the RU is active, and $\alpha_m = 0$ indicates that the RU is in sleep mode. These actions are derived from the output of the DRL models, which guides the energy-efficient operation of the network.

Reward Function

reward $r \in \mathbb{R}$ is designed to balance energy efficiency and user satisfaction, guiding the model towards an optimal operational policy. It is defined as:

$$r = -w_1 \cdot \frac{P_{\text{tot}}}{P_{\max}} - w_2 \cdot \frac{K_{\text{unsat}}}{K}, \tag{4.31}$$

where $P_{\max}$ corresponds to the energy consumption if all RUs are all active. The term K_{unsat} denotes the number of users with data rates below their required thresholds, normalized by the total number of users K. The weights w_1 and w_2 adjust the relative importance of energy efficiency and user sat-

isfaction. Normalization is used in the reward function to ensure that the contributions of energy consumption and user satisfaction are scaled to comparable ranges, preventing dominance by one factor over the other. The original RU activation problem Eq (4.24) can be viewed as minimizing network energy subject to minimum rate constraints. To make the problem tractable within the MDP framework, we adopt a penalty-based relaxation in which QoS violations are incorporated into the reward function. This corresponds to a Lagrangian relaxation of the constrained optimization problem, where the penalty weight w_2 acts as a Lagrange multiplier. Under sufficiently large w_2, the relaxed objective discourages persistent violations and approximates the constrained solution in practice, while maintaining stable gradients for DRL training. During training, multiple weight settings such as $(1, 1)$, $(1, 3)$, and $(1, 10)$ were evaluated. The final choice $(1, 5)$ was selected because it provides a reasonable balance between energy-saving exploration and QoS protection, avoiding both overly conservative and overly aggressive behavior. We note that this tradeoff is observed under the considered simulation configurations. From a user-centric perspective, satisfaction should not be interpreted only as a numerical constraint, but as the perceived ability of the network to provide reliable and continuous service. In this thesis, user satisfaction is represented through the QoS term in the reward function, where a UE is regarded as unsatisfied if its achieved data rate falls below the required threshold. This design reflects the principle that energy saving is valuable only when it does not come at the expense of unacceptable service degradation. Therefore, the reward function penalizes unsatisfied users so that the agent is discouraged from switching off too many RUs purely to reduce power

consumption. The weight w_2 controls the importance of this user satisfaction penalty: a larger w_2 makes the policy more conservative and QoS-protective, while a smaller w_2 allows more aggressive energy-saving behaviour. In this sense, the reward function embeds a philosophical trade-off between infrastructure efficiency and user experience, ensuring that the learned policy seeks energy reduction while maintaining a minimum acceptable level of service for users. Overall, At each time slot t, the agent observes the state s_t, consisting of user data rates, RU loads, RU activation modes, and UE coordinates. Based on this state, the policy $\pi(a_t|s_t)$ outputs an RU activation vector a_t. The environment then evolves according to user mobility and traffic dynamics, producing the next state s_{t+1} and the reward r_t, which jointly reflect both energy consumption and QoS satisfaction.The agent updates the policy to maximize the long-term cumulative reward, thereby learning optimal RU sleep/active decisions.

4.4.2 DQN-Based Model Training

The DQN framework extends traditional Q-learning by employing a deep neural network to approximate the action–value function $Q(s, a; \theta)$, where θ denotes the network parameters. Instead of maintaining a tabular representation, the DQN maps input states to Q-values for all possible actions, enabling decision-making in high-dimensional state spaces. During training, the network parameters are updated to minimize the temporal-difference loss:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\big)^2\Big], \tag{4.32}$$

where θ' is the target network parameter vector, γ is the discount factor, (s, a, r, s') are sampled from the replay buffer D, and the maximization is taken over the discrete action set. The use of a target network and replay experience helps stabilize training by breaking correlations in sequential samples and preventing oscillations.

Building on this framework, we applied DQN to the RU sleep control problem as our initial discrete action solution. In this setting, the agent takes the current network state as input and outputs Q-values for possible RU activation patterns, selecting the action with the highest estimated value. This formulation allows the agent to learn sleep/active scheduling policies without requiring exhaustive optimization at every time step, thus avoiding the computational burden of traditional methods. To investigate how action representation impacts scalability and learning efficiency, we developed two variants: a DQN-Multiple Action (DQNMA), where each action encodes the joint activation state of all RUs, and a DQN-Single Action (DQNSA), where each action controls the state of only one RU at a time. These two designs highlight the trade-off between expressiveness and tractability in DQN-based sleep control.

Multiple-Action DQN (DQNMA)

In the DQNMA model, each action represents a complete sleep/active configuration of the entire RU set. Given M RUs, the total number of possible configurations is 2^M. Instead of representing each configuration as a binary vector, we encode each configuration as an integer index:

$$\mathcal{A}_{\text{multi}} = \{0, 1, \ldots, 2^M - 1\} \tag{4.33}$$

Each action $a \in \mathcal{A}_{\text{multi}}$ is mapped to a binary vector of length M, where the i-th bit determines whether RU i is active (1) or asleep (0). For example, in a system with $M = 6$, the action $a = 42$ corresponds to the binary vector $[1, 0, 1, 0, 1, 0]$, indicating the activation states of all six RUs.

This representation is compact and facilitates Q-value lookup through a single scalar action input. However, the action space still grows exponentially with M, making the Q-table (or output layer of the DQN) increasingly difficult to train. As M increases, it becomes challenging for the agent to explore all possible configurations effectively, leading to slower convergence and suboptimal learning.

Single-Action DQN (DQNSA)

To mitigate the scalability issue inherent in DQNMA, we propose a single-action DQN variant that limits control to one RU per time step. The action space is defined as:

$$\mathcal{A}_{\text{single}} = \{\text{ON}_1, \ldots, \text{ON}_M, \text{OFF}_1, \ldots, \text{OFF}_M\} \tag{4.34}$$

resulting in a total of $2M$ actions. Each action explicitly represents turning ON or OFF a specific RU. This reduces the action space from exponential to linear size, greatly simplifying the learning problem and accelerating convergence. However, the agent's limited per-step control restricts its ability to rapidly adapt to network-wide changes, potentially resulting in suboptimal policies in highly dynamic environments.

4.4.3 TD3 Model Training

Although DQN-based approaches, as we mentioned above, provide effective solutions for discrete RU control, they face significant limitations when applied to large-scale environments with many RUs. In such scenarios, the action space grows exponentially with the number of controllable RUs, making it increasingly difficult for value-based methods to explore and learn optimal policies efficiently. To address these challenges, we adopt a continuous action approach using the TD3 algorithm as shown in Algorithm 3. By modeling the control decisions in a continuous space, TD3 allows the agent to express more accurate preferences over RU activation and significantly reduces the combinatorial complexity of policy learning.

In our adaptation of TD3, the actor network outputs a continuous action vector $\mu_\theta(s) \in [0,1]^M$, where each element represents the activation likelihood of an RU. To enforce discrete sleep/active decisions in the environment, we apply a deterministic thresholding function with Gaussian exploration noise:

$$a_i = f_d(\mu_\theta(s)_i + \eta_i), \quad a_i = \begin{cases} 1 & \text{if } \mu_\theta(s)_i + \eta_i > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4.35}$$

where, $i \in \{1, \ldots, M\}$ indexes the RUs, η_i is a Gaussian noise vector with elements $\eta_i \sim \mathcal{N}(0, \sigma^2)$, adding more exploration in the action decision. $f_d(\cdot)$ denotes the discretization mapping from continuous actor outputs to binary RU activation decisions.This approach enables the policy to explore efficiently in a smoothed action space while still producing discrete activation patterns compatible with RU switching.

The environment includes realistic features such as user mobility, handovers, and RU-specific energy models. Transitions (s, a, r, s') are collected in a replay buffer, and the reward function jointly captures energy efficiency and QoS satisfaction. We use two critic networks Q_{ϕ_1} and Q_{ϕ_2} to compute target Q-values and mitigate overestimation bias:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i}(s', f_d(\mu_{\theta'}(s') + \epsilon)), \tag{4.36}$$

where $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma^2_{\text{target}}))$ provides target smoothing. The actor is updated using the deterministic policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a Q_\phi(s, a)|_{a=\mu_\theta(s)} \cdot \nabla_\theta \mu_\theta(s)]. \tag{4.37}$$

Following TD3's principle of stability, we delay actor and target network updates relative to the critics, and employ soft target updates:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta', \quad \phi_i' \leftarrow \tau\phi_i + (1-\tau)\phi_i'. \tag{4.38}$$

While our TD3-based approach helps the large discrete action space problem by leveraging continuous policy learning and threshold-based execution also provides better results as we show in the simulation section, it does not fully eliminate the scalability limitations of centralized training. As the number of RUs increases, for example in ultra-dense urban environments, the centralized controller continues to encounter significant challenges related to computational complexity, communication overhead, and limited policy generalization. To address these issues, we further explore a federated DRL framework in the next section.

Algorithm 3: TD3-Based Energy-Aware RU Control

Input: Initial network state s_0, actor parameters θ, critic parameters ϕ_1, ϕ_2, replay buffer $\mathcal{D}$

Output: Trained actor network μ_θ for RU sleep scheduling

for *each episode* **do**
- Initialize environment and receive initial state s ;
- **for** *each time step* $t = 1, \ldots, T$ **do**
  - Add exploration noise $\eta_t \sim \mathcal{N}(0, \sigma^2)$;
  - Compute continuous action: $a_c = \mu_\theta(s) + \eta_t$;
  - Discretize: $a = f_d(a_c)$ where $a_i = \mathbb{I}[a_{c,i} > 0.5]$;
  - Execute action a in the environment ;
  - Receive reward r and next state s' ;
  - Store (s, a, r, s') into replay buffer $\mathcal{D}$;
  - Sample a mini-batch (s_j, a_j, r_j, s'_j) from $\mathcal{D}$;
  - Compute target action: $\tilde{a}'_j = f_d(\mu_{\theta'}(s'_j) + \epsilon)$;
  - Compute target Q-value: $y_j = r_j + \gamma \min_{i=1,2} Q_{\phi'_i}(s'_j, \tilde{a}'_j)$;
  - Update critics ϕ_1, ϕ_2 by minimizing: $\mathcal{L}(\phi_i) = \frac{1}{N} \sum_j \left(Q_{\phi_i}(s_j, a_j) - y_j\right)^2$;
  - **if** *every* d *steps* **then**
    - Update actor using deterministic policy gradient ;
    - Soft-update target networks: $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$, $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$;
  - $s \leftarrow s'$

4.5 Simulation Results

4.5.1 Simulation Settings

In this section, we present numerical results to evaluate the performance of the proposed TD3-based model. Our simulations are conducted within an O-RAN environment, where M RUs serve K UEs. The RUs are uniformly spaced across a square service area of dimensions $L \times L$ m^2 illustrated in Fig.4.2. To investigate the impact of network size on model performance, we consider two different service area sizes. This comparison allows us to

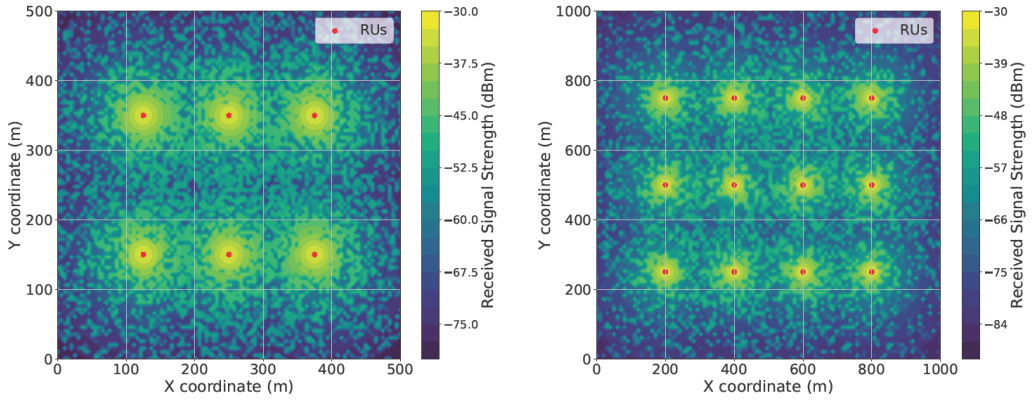


Figure 4.2: illustrates the radio maps of 500 $\times$ 500 m^2 and 1000$\times$ 1000 m^2 area.

assess how performance varies in different spatial environments. It is important to note that the DQN multi-action model is constrained to small service areas due to the exponentially growing action space in larger environments. Consequently, its training feasibility is limited when the network size increases, whereas the other models remain applicable across different service area scales. The UEs move at a speed, denoted as $v = v_{\text{avg}} \pm v_{\text{std}}$, where v_{avg} represents the mean speed and v_{std} denotes the standard deviation of the speed. The actual speed of each UE is randomly selected within this range. Furthermore, we define a movement probability distribution for UEs based on their location to ensure a periodic mobility pattern. This pattern dictates that UEs consistently move from the edge of the service area toward the center and then back to the edge, maintaining a structured and cyclic movement behavior. For the fading channel model, we employ the Urban Microcel (UMi) channel model, considering the conditions Line of Sight (LOS) and Non-Line of Sight (NLOS), expressed as:

For LOS conditions:

$$PL_{m,k}^{\text{LOS}} = \begin{cases} 32.4 + 21\log_{10}(d_{m,k}) + 20\log_{10}(f), & d_{m,k} \leq d_{\text{BP}}, \\ 32.4 + 40\log_{10}(d_{m,k}) + 20\log_{10}(f) - 9.5, & d_{m,k} > d_{\text{BP}}, \end{cases} \tag{4.39}$$

$$d_{\text{BP}} = \frac{4h_{\text{RU}}h_{\text{UE}}f}{c}, \tag{4.40}$$

where d_{BP} is the breakpoint distance, $PL_{m,k}$ represents the path loss, which follows the UMi path loss model.

For NLOS conditions:

$$\begin{aligned} PL_{m,k}^{\text{NLOS}} = 35.3 &+ 22.4\log_{10}(d_{m,k}) \\ &+ 21.3\log_{10}(f) - 0.3(h_{\text{UE}} - 1.5), \end{aligned} \tag{4.41}$$

where $d_{m,k}$ denotes the distance between the RU m and the UE k, f is the carrier frequency in GHz, and h_{UE} is the height of the UE in meters.

The LOS probability is determined by:

$$P_{\text{LOS}} = \min\left(\frac{18}{d_{m,k}}, 1\right)\left(1 - e^{-d_{m,k}/36}\right) + e^{-d_{m,k}/36}. \tag{4.42}$$

The NLOS probability is then given by $P_{\text{NLOS}} = 1 - P_{\text{LOS}}$.

During the training process, each episode consists of 200 time steps, with each time step corresponding to 1 second. At every time step, the agent determines the sleep mode status of the RUs, making operational decisions accordingly. The network energy consumption is then evaluated based on these decisions. As a result, the total simulation duration for each episode

Parameters	Value
Carrier frequency (f)	2 GHz
RU height (h_{RU})	15 m
UE height (h_{UE})	1.7 m
Network size (L)	[500,1000] m
Number of RUs (M)	[6, 12]
Number of UEs (K)	[10 - 80]
Minimum Data rate requirement ($R_{\min}$)	3 Mbps
Noise power (σ_n^2)	-174 dBm/Hz
Power amplifier (η)	0.5
Average speed of UE (v_{avg})	2 m/s
Standard deviation of UE speed (v_{std})	0.5 m/s
Fix active mode RU power(P^{active})	20 W
Fix sleep mode RU power (P^{sleep})	5 W
Maximum transmission power (P_{TX})	1 W / 30dBm
Mode transition power (P^{trans})	3 W

Table 4.1: Simulation Parameters

	Layer1	Layer2	Layer3	Layer4	Output Layer
Actor	BN+relu 512	relu 256	relu 128	None	sigmoid M
Critic	relu 512	relu 256	relu 128	None	linear 1
DQNSA	relu 512	relu 384	relu 256	relu 128	linear M^2
DQNMA	relu 512	relu 384	relu 256	relu 128	linear $2M$

Table 4.2: Network Configurations

amounts to 200 seconds. In the TD3 model, both the actor and critic networks consist of four fully connected layers. The configuration of activation functions and layer sizes are detailed in Table 4.1. Additionally, Batch Normalization (BN) is incorporated to enhance training stability. For the DQN models, both networks comprise five fully connected layers, with their detailed architectures also provided in Table 4.2.

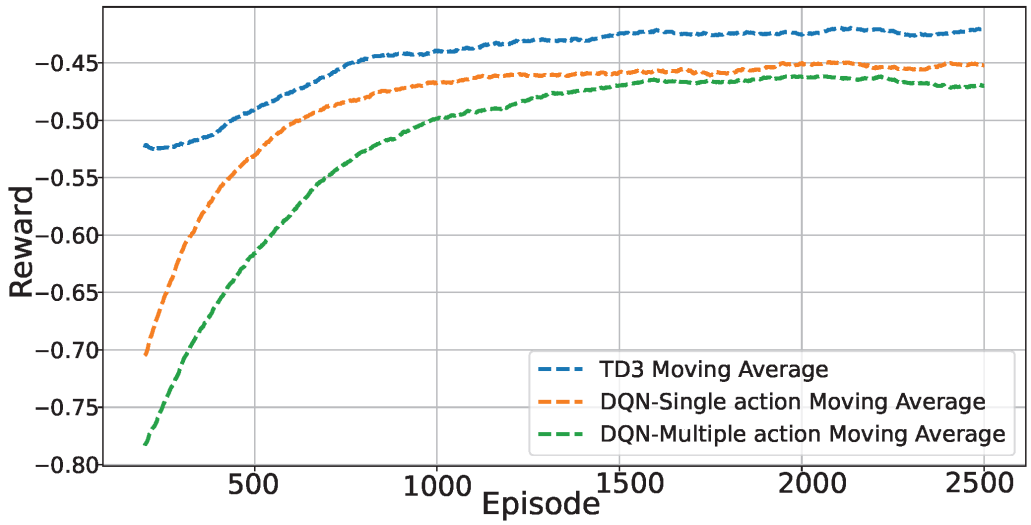


Figure 4.3: The rewards of TD3, DQNSA and DQNMA in 500 × 500 m^2 scenario

In the subsequent section, we present a comparative analysis of the proposed TD3 model against three baseline techniques for RU sleep mode management. The first baseline, termed "All Active Mode," serves as a reference scenario where all RUs remain continuously active. The second baseline is the DQNSA model, which dynamically switches the sleep mode status of only one RU per time slot. The third baseline, referred to as the DQNMA model, simultaneously adjusts the sleep modes for all RUs within the network at each time interval.

4.5.2 Simulation Results

In Fig.4.3 illustrates the reward performance over 2500 episodes for the TD3, DQNSA and DQNMA models in 500 × 500 m^2 area. Notably, the TD3 model consistently achieves the highest reward performance, demonstrating superior stability and convergence compare to both DQN models. The DQNSA model exhibits moderate performance with noticeable fluctuations, while the

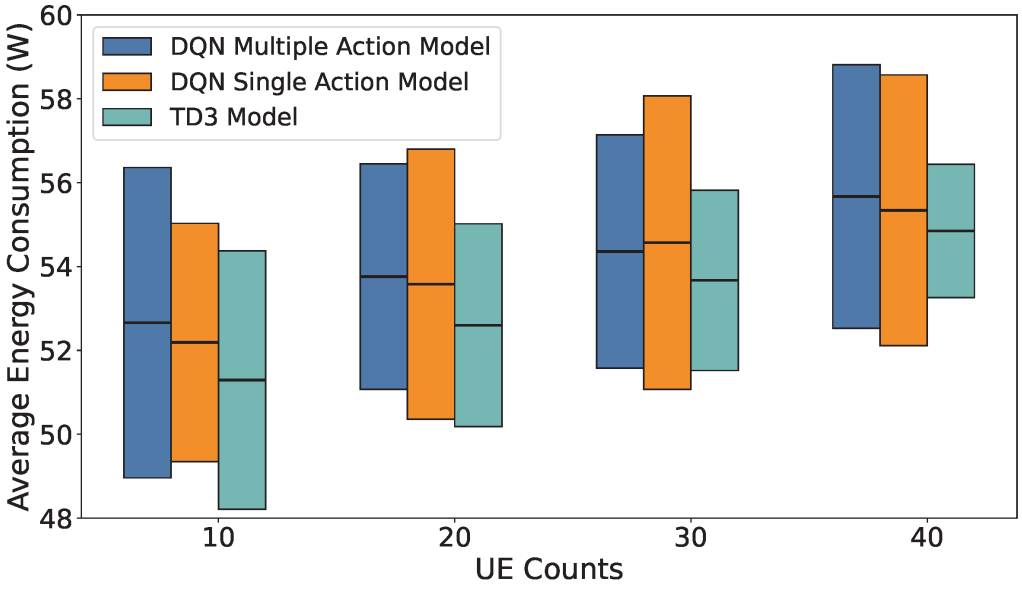


Figure 4.4: The average energy consumption in 500 $\times$ 500 m^2 scenario among 10 to 40 UEs with DQNMA, DQNSA and TD3 models.

DQNMA model shows the lowest reward, indicating that increasing the complexity of actions within the DQN architecture does not necessarily translate into better reward outcomes in this scenario. In Fig.4.4 indicates the energy consumption performance in 500 $\times$ 500 m^2 area for varying UE counts ranging from 10 to 40, is still compared among the previous three models. The figure distinctly shows the average energy consumption, indicated by the black line in the center of each bar, while the length of each bar represents the variance in energy consumption. With a total of 6 RUs installed in this area, the maximum possible energy consumption reaches 126w when all RUs operate in active mode. The results clearly demonstrate that all three models achieve significant energy savings, exceeding 50% compared to the theoretical maximum. Specifically, the TD3 model achieving up to an additional 6% energy saving compared to the two DQN-based models. Conversely, the

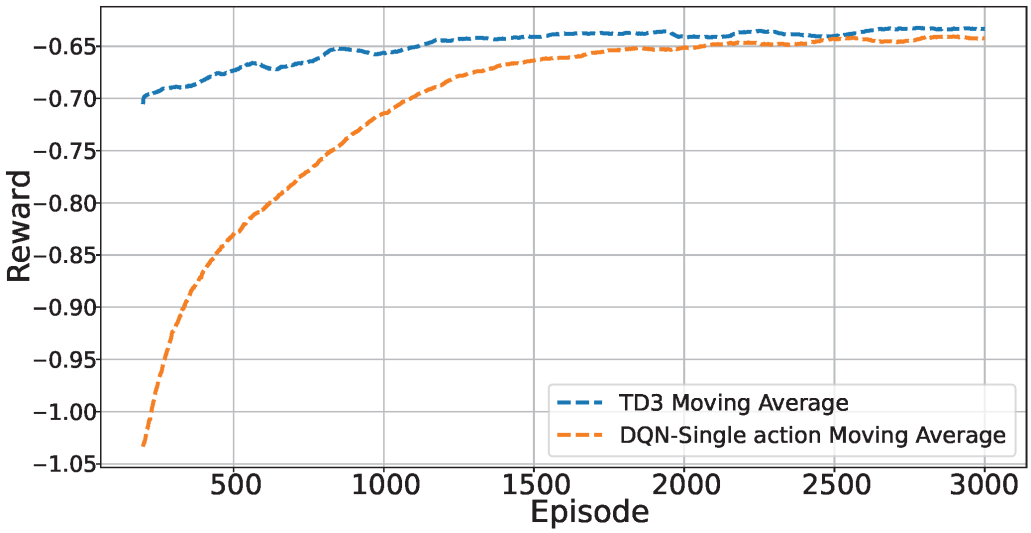


Figure 4.5: The rewards of TD3 and DQNSA model in 1000 × 1000 m^2 scenario

DQNMA model demonstrates the highest energy consumption, underscoring potential inefficiencies associated with more complex action spaces.

In Fig.4.5 shows the reward performance over an extended simulation environment from 500 × 500 m^2 area to 1000m^2 area. But only compare the TD3 model against the DQNSA model due to the excessive expansion of DQNMA action space (e.g. 2^{12}). In the results, the TD3 model still achieves higher and stable reward levels throughout all episodes. Furthermore, the TD3 model demonstrates faster and more efficient convergence toward optimal solutions. The Fig.4.6 illustrates the energy consumption between the TD3 and DQNSA model, covering UE counts from 20 to 80. Numerically, the TD3 model maintains lower energy consumption, approximately 144w at 20 UEs and up to around 152w at 80 UEs, significantly below the theoretical maximum consumption of 252 watts (more than 40%) with all 12 RUs active. However, the TD3 model can also achieve 5% over the DQNSA mode, further emphasizing the TD3 model in a complex and large-scale environment.

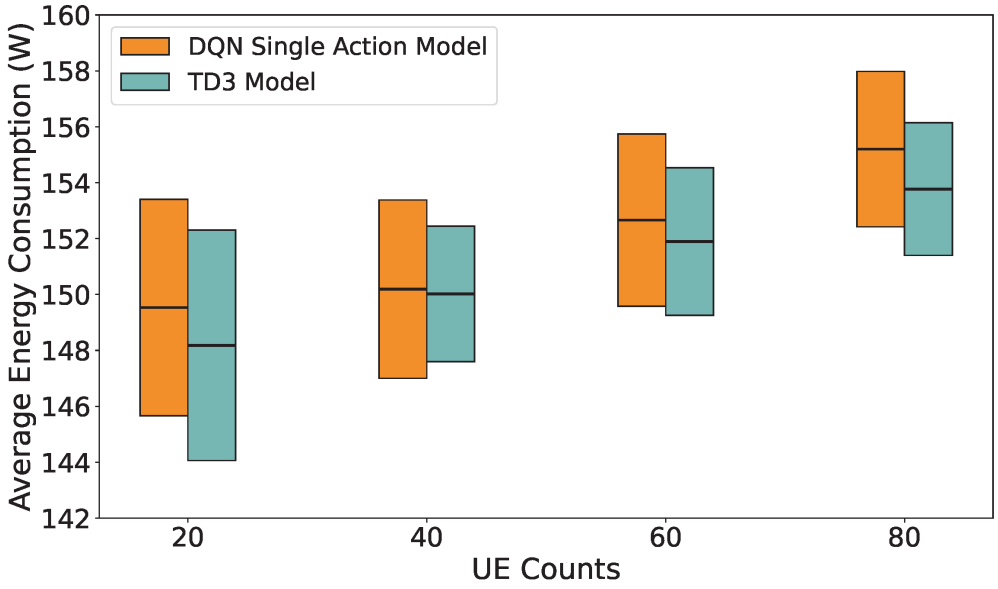


Figure 4.6: The average energy consumption among 20 to 80 UEs in 1000 × 1000 m^2 scenario with TD3 and DQNSA models.

4.6 Summary

This chapter proposed an energy-efficient framework for controlling RU sleep modes in O-RAN using advanced DRL methods. Proposed TD3 algorithm was implemented to effectively handle the continuous action space inherent to RU state decisions, directly addressing the limitations encountered by discrete-action methods like DQN-based approaches. In comparison, the proposed TD3 algorithm demonstrated superior performance, consistently achieving higher stability, faster convergence, and greater energy efficiency. Numerical results clearly indicated that TD3 delivered up to 6% additional energy savings over the baseline DQN methods, particularly highlighting its scalability and effectiveness in larger and more dynamic network scenarios. Future research directions will focus on enhancing this TD3-based approach through FL. This future work aims to leverage FL principles to further

enhance scalability.

5 Federated Reinforcement Learning for Scalable O-RAN Optimization

5.1 Introduction

In Chapter 4, a centralized DRL framework based on the TD3 and DQN algorithm was proposed to optimize RU sleep mode decisions in an O-RAN environment. By leveraging a continuous action space, the TD3-based approach effectively mitigates the action space explosion problem inherent in traditional discrete-action methods such as Deep Q-Networks (DQN), achieving significant improvements in both energy savings and convergence stability.

Despite these advantages, centralized DRL approaches face inherent limitations when applied to large-scale and geographically distributed O-RAN deployments. As the number of RUs increases, centralized training requires aggregating large volumes of data and performing computation at a single controller, leading to scalability bottlenecks and increased communication overhead. Furthermore, such approaches do not fully exploit the hierarchical and distributed intelligence enabled by the O-RAN architecture, where control and learning functionalities are naturally separated between the Near-RT RIC and the Non-RT RIC

To address these challenges, this chapter extends the centralized DRL

framework into a scalable FRL-based architecture tailored for O-RAN systems. In the proposed framework, multiple local agents deployed as xApps within Near-RT RICs independently learn RU control policies based on regional observations, while a global model is coordinated by the Non-RT RIC through periodic parameter aggregation. This design significantly reduces communication overhead, enhances scalability, and improves policy generalization across heterogeneous network conditions.

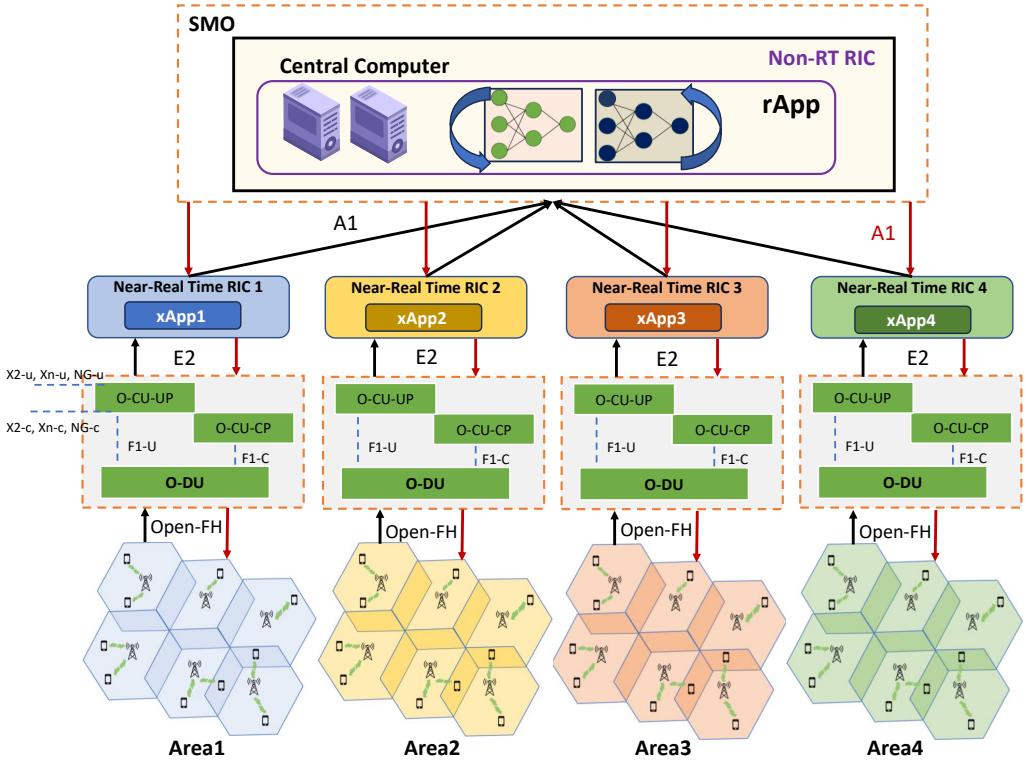


Figure 5.1: Illustration of the O-RAN architecture incorporating both O-RAN-defined and 3GPP-standard interfaces. Solid lines represent O-RAN interfaces (e.g., E2, A1, O1, Open FH), while dashed lines indicate 3GPP interfaces.The system spans four geographical areas, each with its own O-DU and O-CU components. Near-RT RICs operate on a per-area basis, deploying multiple xApps. A centralized Non-RT RIC performs global policy aggregation and training coordination using A1 interface.

5.2 Limitations of Centralized DRL

While centralized DRL frameworks have demonstrated strong performance in optimizing energy efficiency for wireless networks, their applicability in large-scale O-RAN systems is constrained by several fundamental limitations.

First, centralized training introduces significant computational bottlenecks as the network size increases. In dense deployments, the number of controllable RUs grows rapidly, resulting in a high-dimensional state and action space[9]. Although continuous-action algorithms such as TD3 alleviate the combinatorial explosion associated with discrete methods, the overall training complexity remains substantial, particularly when real-time decision-making is required.

Second, centralized approaches rely on collecting data from all network elements at a single controller, which leads to considerable communication overhead. In practical O-RAN deployments, where RUs and DUs are geographically distributed, frequent data exchange can introduce latency and place a heavy burden on fronthaul and backhaul links. This issue becomes more pronounced in ultra-dense networks with high user mobility and dynamic traffic patterns.

Third, centralized DRL does not align well with the inherently distributed and hierarchical nature of O-RAN architectures. O-RAN introduces a clear separation between long-term optimization and real-time control through the Non-RT RIC and Near-RT RIC, respectively [19]. Centralized learning frameworks fail to fully leverage this structure, limiting their ability to scale efficiently across multiple regions and adapt to localized network dynamics.

Finally, centralized learning suffers from limited generalization in hetero-

geneous environments. When network conditions vary significantly across different geographical regions, a single global policy may not capture region-specific characteristics effectively, leading to suboptimal performance. This limitation highlights the need for distributed learning mechanisms that can adapt locally while maintaining global coordination.

To overcome these challenges, federated learning has emerged as a promising paradigm for distributed model training. By allowing local agents to train models independently and share only model parameters instead of raw data, federated learning significantly reduces communication overhead and enhances scalability [72]. In the context of DRL, federated approaches enable multiple agents to collaboratively learn a shared policy while preserving local adaptability, making them well-suited for large-scale O-RAN deployments.

5.3 Chapter Contributions

The main contributions of this chapter are summarized as follows:

- Building upon the centralized DRL framework, we extend the RU sleep control problem to large-scale O-RAN deployments by introducing a distributed learning architecture.
- We propose a FDRL framework in which multiple local agents (xApps) operate within Near-RT RICs, while a global model is coordinated by the Non-RT RIC through parameter aggregation. This design aligns naturally with the hierarchical structure of O-RAN.
- By exchanging model parameters instead of raw data, the proposed framework significantly reduces communication overhead and improves

scalability in geographically distributed networks.

- The FL paradigm enhances convergence speed and reduces training energy consumption by leveraging parallel learning across multiple regions.

The remainder of this chapter is organized as follows. Section 5.4 introduces the extended multi-region system model and problem formulation and the hierarchical O-RAN architecture. Section 5.5 presents the proposed federated DRL framework, including local training and global aggregation mechanisms. Section 5.6 provides the simulation setup for large-scale scenarios and analyzes the performance results, highlighting scalability and convergence improvements. Finally, Section 5.7 concludes the chapter.

5.4 System Extension for Multi-Region O-RAN

The system model considered in this chapter builds upon the unified framework introduced in Chapter 3 and the centralized setting described in Chapter 4 Section 4.3. In order to enable scalable and distributed learning, the system is extended to a multi-region O-RAN deployment with hierarchical control.

Specifically, the network is partitioned into multiple geographical regions as illustrated at Fig. 5.1, each consisting of a set of RUs and UEs. Each region is managed by an independent Near-RT RIC, which hosts local xApps responsible for real-time control decisions. These xApps implement DRL agents that interact with the local environment, observe network states, and

determine the activation or sleep modes of RUs within their respective regions.

At a higher level, a centralized Non-RT RIC is introduced to coordinate global learning and policy optimization. Instead of directly controlling RUs, the Non-RT RIC aggregates model parameters from all regional agents and redistributes an updated global model. This hierarchical control structure aligns with the O-RAN architecture, where the Non-RT RIC operates on a longer timescale for policy management and optimization, while the Near-RT RIC performs latency-sensitive control tasks.

From a system perspective, each region can be modeled as an independent MDP with local state observations, including user distribution, traffic load, and RU activation states. The action space and state representation follow those defined in Chapter 4 Section 4.4.1, ensuring consistency in decision-making across centralized and distributed settings.

5.4.1 Problem Formulation in Federated Setting

The optimization objective in this chapter remains consistent with that defined in Chapter 4 Section 4.3.1, namely, to minimize the total network energy consumption while satisfying the QoS constraints of all UEs. Formally, the problem can be expressed as a long-term cumulative energy minimization task subject to data rate and resource allocation constraints.

Unlike the centralized formulation, where a single agent observes the global network state and directly optimizes the system-wide objective, the federated setting decomposes the learning process across multiple regions. Each regional agent optimizes a local objective based on its own observations,

while contributing to the improvement of a shared global policy through periodic model aggregation.

Let π_i denote the policy learned by the agent in region i, and π_g denote the global policy maintained at the Non-RT RIC. Each local agent interacts with its environment to maximize the expected cumulative reward:

$$J_i(\pi_i) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_i^t\right], \tag{5.1}$$

In each region, the reward follows the same energy-QoS formulation used in Chapter 4. For region i, the reward at time t is expressed as

$$r_i^t = -w_1 * \frac{P_{\text{tot},i}^t}{P_{\text{max},i}} - w_2 * \frac{K_{\text{unsat},i}^t}{K_i}, \tag{5.2}$$

where $P_{\text{tot},i}^t$ is the total energy consumption of region i at time t, $P_{\text{max},i}$ is the energy consumption when all RUs in region i are active, $K_{\text{unsat},i}^t$ is the number of unsatisfied UEs whose achieved data rates are below their required thresholds, and K_i is the total number of UEs in region i. The weights w_1 and w_2 balance the relative importance of energy saving and QoS satisfaction.

The global objective can then be interpreted as the aggregation of regional objectives:

$$J_g = \sum_{i=1}^{N} \omega_i J_i(\pi_i), \tag{5.3}$$

where ω_i represents the relative importance or weighting factor of each region.

Through federated learning, local agents periodically upload their model parameters to the Non-RT RIC, where a global aggregation step is performed to obtain an updated global policy π_g. This policy is then redistributed to

all regions, enabling coordinated learning while preserving local adaptability.

It is important to note that, while the learning paradigm changes from centralized to distributed, the underlying optimization problem, system dynamics, and reward design remain unchanged. This ensures that the proposed federated DRL framework is directly comparable to the centralized approach presented in Chapter 4, while significantly improving scalability and deployment feasibility in large-scale O-RAN systems.

5.5 Federated DRL for Scalable RU Control in O-RAN

The modular and disaggregated architecture of O-RAN makes it a natural fit for federated DRL, particularly in scenarios that involve large-scale and geographically distributed RU deployments. In O-RAN, control functionality is split across multiple layers, such as xApps operating at the Near-RT RIC and rApps at the Non-RT RIC, communicating via standardized open interfaces. This structure aligns seamlessly with the federated learning paradigm, where learning agents (e.g., xApps) act as local clients that train policies independently using site-specific data and periodically synchronize with a central coordinator (e.g., an rApp) to build a global policy model [73, 74].

Adopting federated DRL in this context offers significant scalability advantages over centralized DRL. As network size and RU density increase, centralized learning suffers from exploding state and action spaces, as well as communication bottlenecks and slow convergence. By distributing the learning process across O-RAN components, federated DRL reduces com-

putational and communication loads at any single point [75, 76], while also enabling site-level policy customization. Moreover, the reuse of O-RAN's open interfaces allows efficient and standards-compliant integration of FL workflows into real network deployments. An overview of the proposed federated DRL framework within the O-RAN architecture is depicted in Fig. **??**. In this design, each O-RAN region is equipped with a Near-RT RIC hosting local DRL agents that interact with the underlying RUs and UEs to make real-time sleep/active decisions. These agents are trained locally using region-specific traffic and mobility dynamics, thereby capturing heterogeneous environmental characteristics without requiring raw data exchange. Periodically, the locally trained models are uploaded to the Non-RT RIC, which serves as the global aggregator. The Non-RT RIC integrates model updates from multiple regions to construct a global policy, which is then redistributed back to the Near-RT RICs. This hierarchical workflow aligns naturally with the functional split of O-RAN: the Near-RT RIC ensures responsiveness to fast-varying network states, while the Non-RT RIC provides global coordination, scalability, and policy generalization across distributed deployments.

5.5.1 Federated DRL Training and Global Aggregation

In our federated DRL framework, each agent corresponds to a distinct geographical region, consisting of multiple RUs and UEs. The training process begins with the aggregator distributing an initialized global model to all participating agents. This model includes either Q-network weights for DQN agents or actor–critic parameters for TD3 agents, depending on the

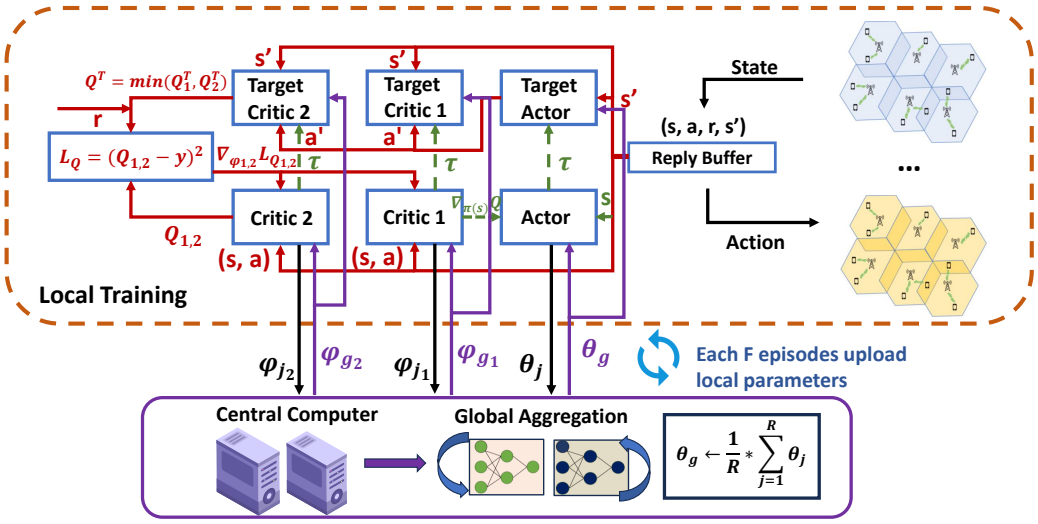


Figure 5.2: Illustration the workflow of Fed-TD3 across distributed agents and a aggregator. Red solid lines indicate local critic training based on temporal-difference loss; green dashed lines represent actor updates guided by critic gradients and soft update target network; black and purple solid lines denote the periodic aggregation and redistribution of actor and critic parameters via the aggregator. Each agent operates within its own local environment and contributes to a globally coordinated policy through federated learning.

algorithm. Fig. 5.2 illustrates the entire Fed-TD3 training process as a representation. After receiving the global model, each agent independently interacts with its local environment, collecting state–action–reward transitions and updating its DRL model based on region-specific user mobility and traffic dynamics. This local training proceeds over a fixed number of episodes F. Once the local training phase is completed, each agent uploads selected model parameters to the aggregator. The aggregator then performs model aggregation to produce an updated global model. This aggregation strategy varies depending on the type of DRL model used and will be detailed in the following section. The updated global model is subsequently redistributed to all agents for the next round of training. This iterative cycle of local

Algorithm 4: Federated TD3 Training Process

Require: Multi-region environment with R regions, local TD3 agents $\{A_i\}_{i=1}^{R}$, aggregation frequency F and TD3 hyperparameters.

Initialization:
for $i = 1$ **to** R **do**
    Initialize local agent A_i with its actor, critics, and target networks.
end for
Initialize the global model parameters θ_{global}, $\phi_{\text{global\_1}}$ and $\phi_{\text{global\_2}}$ with random value. Establish the connection between local agents and a global server.
Training Loop:
for episode $= 1$ **to** N_{eps} **do**
    for $j = 1$ **to** R **do**
        Initialize state $s_0^{(i)}$.
        for $t = 0$ **to** $T - 1$ **do**
            Agent A_i selects an action $a_t^{(i)}$
            Execute actions $a_t^{(i)}$ in the environment.
            Obtain next states $s_{t+1}^{(i)}$ and rewards $r_t^{(i)}$.
            Agent A_i stores transition $(s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)})$.
            Agent A_i performs a local TD3 update.
            if $(t + 1) \bmod F = 0$ **then**
                Global Server Aggregation (FedAvg). Agent A_i sends θ_i, ϕ_{i_1} and ϕ_{i_2} to the global server.
            end if
            Update state: $s_t^{(i)} \leftarrow s_{t+1}^{(i)}$.
        end for
    end for
end for
Global Server Aggregation (FedAvg):
the The global server receives local parameters from all agents.
compute the weighted average of actor and critic model update:
$\theta_{\text{avg}} = \sum_j (\omega_j \cdot \theta_j / \sum_j \omega_j)$
$\phi_{\text{avg\_}i} = \sum_j (\omega_{j\_i} \cdot \phi_{j\_i} / \sum_{j\_i} \omega_{j\_i}), \quad i = 1, 2$. Where ω_i are weight factors
Global Model Update:
Update the global actor and critic model parameters:
$\theta_{\text{global}} = \theta_{\text{avg}}, \quad \phi_{\text{global\_}i} = \phi_{\text{avg\_}i}, \quad i = 1, 2,$
$\theta'_{\text{global}} = \theta_{\text{global}}, \quad \phi'_{\text{global\_}i} = \phi_{\text{global\_}i}, \quad i = 1, 2.$
Global Distribution: For each region j, update local parameters: $\theta_j \leftarrow \theta_{\text{global}}$,
$\phi_{j\_i} \leftarrow \phi_{\text{global\_}i} \quad i = 1, 2,$
$\theta'_j \leftarrow \theta'_{\text{global}}, \phi'_{j\_i} \leftarrow \phi'_{\text{global\_}i}$.

update and global synchronization continues until convergence. By leveraging this decentralized optimization strategy, our framework ensures that agents collaboratively learn a generalizable policy while preserving privacy and significantly reducing communication overhead.

Depending on the underlying DRL algorithm, we employ two distinct parameter-sharing strategies to support both value-based and policy-based learning. The complete Fed-TD3 training process is summarized in Algo-

rithm 4. For agents using the DQN architecture with discrete action spaces, each agent performs Q-learning on local environment and periodically uploads its Q-network to the server. The global model is computed by averaging the Q-values across all agents:

$$Q_{\text{global}}(s,a) = \frac{1}{R}\sum_{j=1}^{R} Q_j(s,a), \quad \forall s,a, \tag{5.4}$$

$$Q_j(s,a) \leftarrow Q_{\text{global}}(s,a), \quad \forall s,a, \tag{5.5}$$

where $Q_j(s,a)$ represents the Q-network of agent j, and R is the number of agents. The aggregated global model $Q_{\text{global}}(s,a)$ is then broadcast to all agents for subsequent training rounds.

For agents using the TD3 algorithm in the continuous action space, each agent maintains an actor–critic architecture. The global policy seeks to learn $|\mathcal{S}|\times|\mathcal{A}|$ table, $\pi_{global}(a|s)$. After several local update episodes, the policy network which is the actor network parameters is uploaded to the server for aggregation:

$$\pi_{\text{global}}(a|s) = \frac{1}{R}\sum_{j=1}^{R} \pi_j(a|s), \quad \forall s,a, \tag{5.6}$$

$$\pi_j(a|s) \leftarrow \pi_{\text{global}}(a|s), \quad \forall s,a. \tag{5.7}$$

Although the standard actor–critic framework typically aggregates only the actor to ensure decentralized critic adaptation, in our design we also upload and aggregate the two critic networks across agents. This additional aggregation improves training stability and enhances the generalizability of the learned value functions. After each global aggregation step, both the actor and critic target networks are updated, as detailed in Algorithm 4. In our

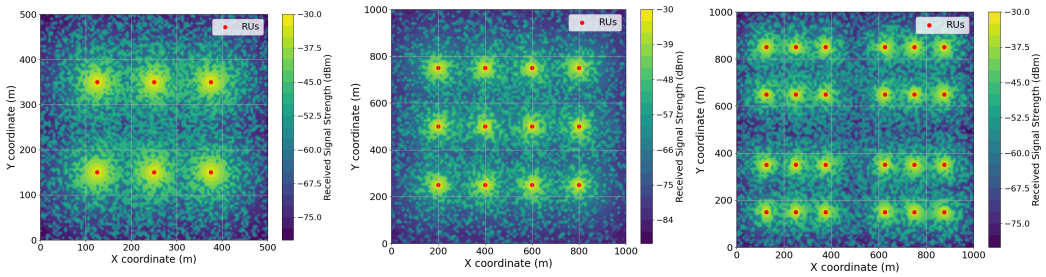


Figure 5.3: The left plot shows a radio map of single 500 m×500 m area used for centralized training and evaluation. The right plot depicts a 1000 m×1000 m area composed of four such subregions, representing a composite environment for centralized training and inference. This comparison setting is used to evaluate the generalization capability of the global model trained via federated reinforcement learning versus a centralized model trained on the combined area. The middle plot presents a single large-scale 1000 m×1000 m environment used to evaluate model performance under centralized DRL training.

scenario, the areas share a homogeneous RU activation task and similar traffic patterns, which reduces the degree of heterogeneity. In the future works we will investigate heterogeneous environments that cause further challenges. In this work we mainly want to show that Federated learning is a scalable AI solution for O-RAN networks.

5.6 Simulation Results

To evaluate the effectiveness of the proposed energy savings framework, we perform simulations across a range of network scenarios and baseline models. Our evaluation includes centralized and federated learning environments to assess performance in terms of reward, energy consumption, and scalability.

5.6.1 Simulation Settings

In our simulations, each scenario is modeled as an O-RAN environment where M RUs serve K UEs within a square area of size $L \times L\,\mathrm{m}^2$. In practical O-RAN deployments, the Near-RT RIC is commonly deployed in edge cloud or regional edge data centers with latency budgets between 10 ms and 1 s. In our system model, the inference of the DRL agent runs inside the Near-RT RIC. The model training or federated aggregation may take place at the same edge location. This placement ensures that the decision-making latency requirements of the xApp are satisfied. As illustrated in Fig. 5.3, three layouts are considered: (i) a 500 m×500 m area with 6 RUs, used both as the centralized DRL testbed and as the local region in FL; (ii) a 1000 m×1000 m area with 12 RUs, representing a large-scale single region; (iii) a 1000 m×1000 m composite area with 24 RUs, formed by four independent 500 m×500 m subregions, used to examine the scalability of FL.

The UEs move at a constant speed $v = v_{\text{avg}} \pm v_{\text{std}}$ (mean $v_{\text{avg}} = 2\,\mathrm{m/s}$, standard deviation $v_{\text{std}} = 0.5\,\mathrm{m/s}$), with individual speeds randomly drawn from this range. A periodic mobility pattern is applied, where UEs travel from the service area edge toward the center and back, forming a structured cyclic movement. The wireless channel adopts the 3GPP UMi model [77] with both LOS and NLOS conditions. Other system parameters, including carrier frequency, antenna heights, bandwidth, and RU power consumption values, follow Table 6.1.

To investigate scalability in large-scale or distributed deployments, each of these approaches is further extended to a federated learning setting, resulting in the Fed-DQNSA, Fed-DQNMA, and Fed-TD3 models. In the fed-

Table 5.1: Simulation Parameters

Parameter	**Value**
TD3 Actor learning rate (α_π)	0.0001
TD3 Critic learning rate (α_Q)	0.001
DQN learning rate (α)	0.0001
Mini-batch size (B)	128
Replay memory size ($\mathcal{R}$)	50000
Discount factor (γ)	0.99
Training episodes (N_{eps})	2000
Random seed	42
Soft update coefficient (τ)	0.01
Carrier frequency (f)	2 GHz
RU height (h_{RU})	15 m
UE height (h_{UE})	1.7 m
Network size (L)	[500, 1000] m
Number of RUs (M)	[6, 12, 24]
Number of UEs (K)	[20–80]
Number of local regions (R)	8
Minimum data rate requirement ($R_{\min}$)	3 Mbps
Noise power (σ_n^2)	-174 dBm/Hz
Power amplifier efficiency (η)	0.5
Average UE speed (v_{avg})	2 m/s
Std. deviation of UE speed (v_{std})	0.5 m/s
Active mode RU power (P^{active})	20 W
Sleep mode RU power (P^{sleep})	5 W
Maximum transmission power (P_{TX})	1 W / 30 dBm
Mode transition power (P^{trans})	3 W
Reward weights (w_1, w_2)	1, 5

erated setting, agents are trained on separate local regions and periodically aggregated into a global model, whereas the centralized setting relies on a single combined environment. This design enables a systematic comparison between centralized and federated paradigms across different DRL architectures.

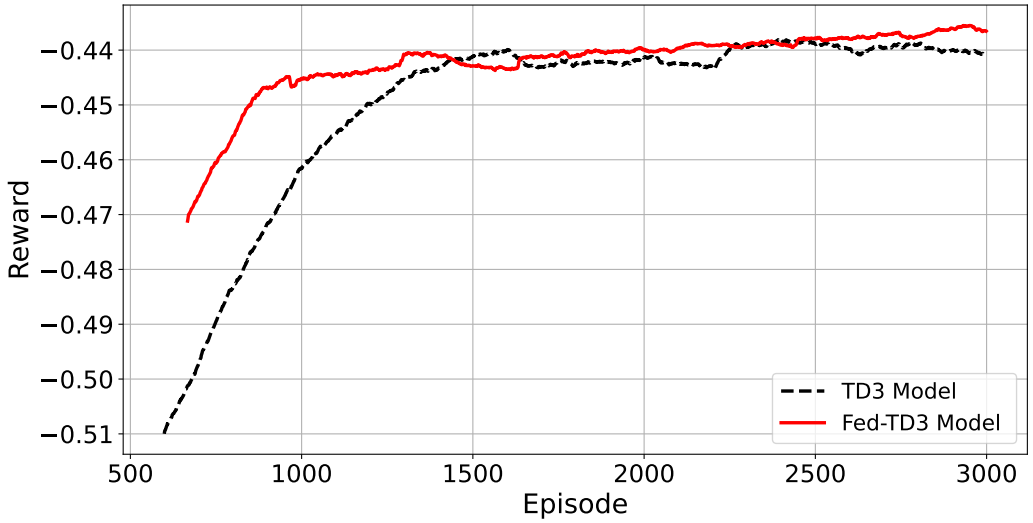


Figure 5.4: The rewards and convergence speed of TD3 and Fed-TD3 in 500 m×500 m area with 6 RUs and 20 UEs.

5.6.2 Numerical Results

As illustrated in Fig. 5.4, the training reward curves highlight clear advantages of the federated framework. To statistically validate the convergence behavior, we computed the convergence episode for five independent runs of both TD3 and Fed-TD3 and performed a Welch two-sample t-test. Fed-TD3 converges in 858.8 ± 127.4 episodes on average, while TD3 requires 1421.4 ± 322.6 episodes. Fed-TD3 achieves 40% reduction in convergence episodes compared with TD3. The t-test yields $p = 0.0309 < 0.05$, indicating that the faster convergence of Fed-TD3 is statistically significant at the 95% confidence level. The observed behavior is interpreted as an empirical outcome of the proposed multi-area hierarchical learning setup. The improved learning dynamics can be attributed to the aggregation of diverse policy updates across regions, which enhances generalization and stabilizes the learning process. Moreover, Fed-TD3 attains a higher final reward com-

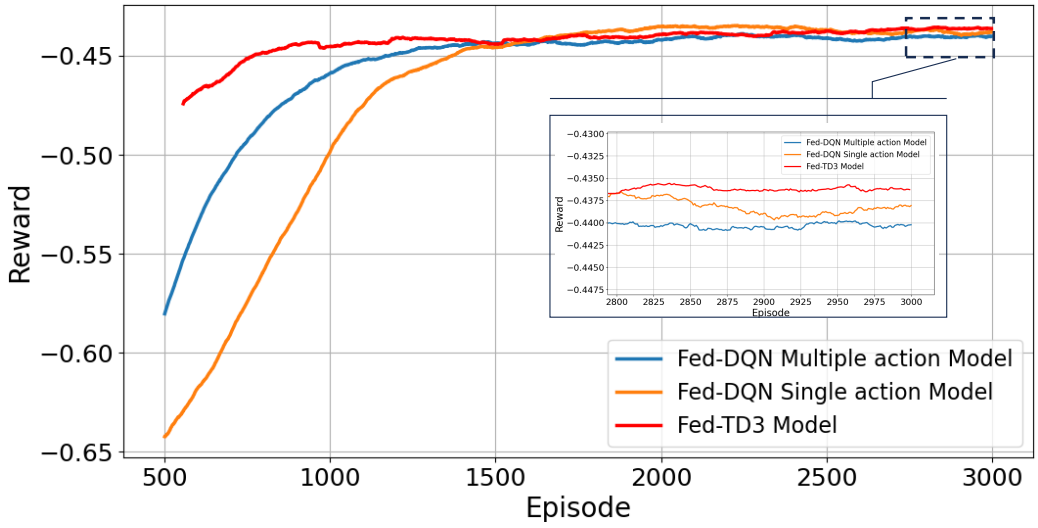


Figure 5.5: The rewards and convergence speed of Fed-TD3, Fed-DQNSA and Fed-DQNMA model in 500 m×500 m area with 6RUs and 20UEs.

pared to centralized TD3, indicating superior long-term performance in balancing energy savings and service quality.

Fig. 5.5 compares the training reward curves of three federated DRL models: Fed-TD3, Fed-DQNMA, and Fed-DQNSA. Both federated DQN variants exhibit significant improvements over their non-federated counterparts, with approximately a 10% increase in final average reward. This demonstrates that federated learning effectively enhances the generalization and learning efficiency of value-based methods even under discrete action settings. Notably, the final rewards achieved by Fed-DQNMA and Fed-DQNSA closely approach that of Fed-TD3, suggesting that with appropriate aggregation, discrete-action methods can benefit substantially from distributed training. However, the convergence speed of Fed-DQNMA and Fed-DQNSA remains noticeably slower than Fed-TD3. While Fed-TD3 stabilizes after approximately 900 episodes, the federated DQN models require around 1,300

episodes to reach similar stability, indicating a relative slowdown of about 30%. This gap reflects the inherent limitations of Q-learning in handling large action spaces compared to policy-based approaches.

Fig. 5.6 compares the average energy consumption between federated and centralized reinforcement learning approaches. The federated models (e.g. Fed-DQNMA, Fed-DQNSA, and Fed-TD3) are first trained across four geographically separated 500 m×500 m regions. After convergence, the resulting global models are independently evaluated in each of the original training regions, and the total test energy consumption is obtained by summing across all four areas. In contrast, the centralized models (DQN and TD3) are trained on a merged 1000 m×1000 m, environment composed of the same four subregions, with user distributions and mobility constrained within each 500 m×500 m area, as illustrated on the third figure of Fig. 5.3. This setting ensures that both federated and centralized models are tested on identical underlying environments, allowing a fair comparison of generalization and efficiency. The results show that federated models achieve comparable or lower overall energy consumption than their centralized counterparts, despite being trained without direct access to globally aggregated data. In particular, Fed-DQNMA and Fed-DQNSA demonstrate strong energy-saving behavior, while Fed-TD3 incurs slightly higher energy usage. This difference arises because Fed-TD3 prioritizes satisfying the QoS requirements of all users, leading to fewer RUs being turned off during testing. In contrast, the fed-DQN models are more aggressive in RU deactivation, sacrificing service quality for a greater reduction in energy consumption. Nevertheless, Fed-TD3 achieves higher average rewards, indicating a better balance between

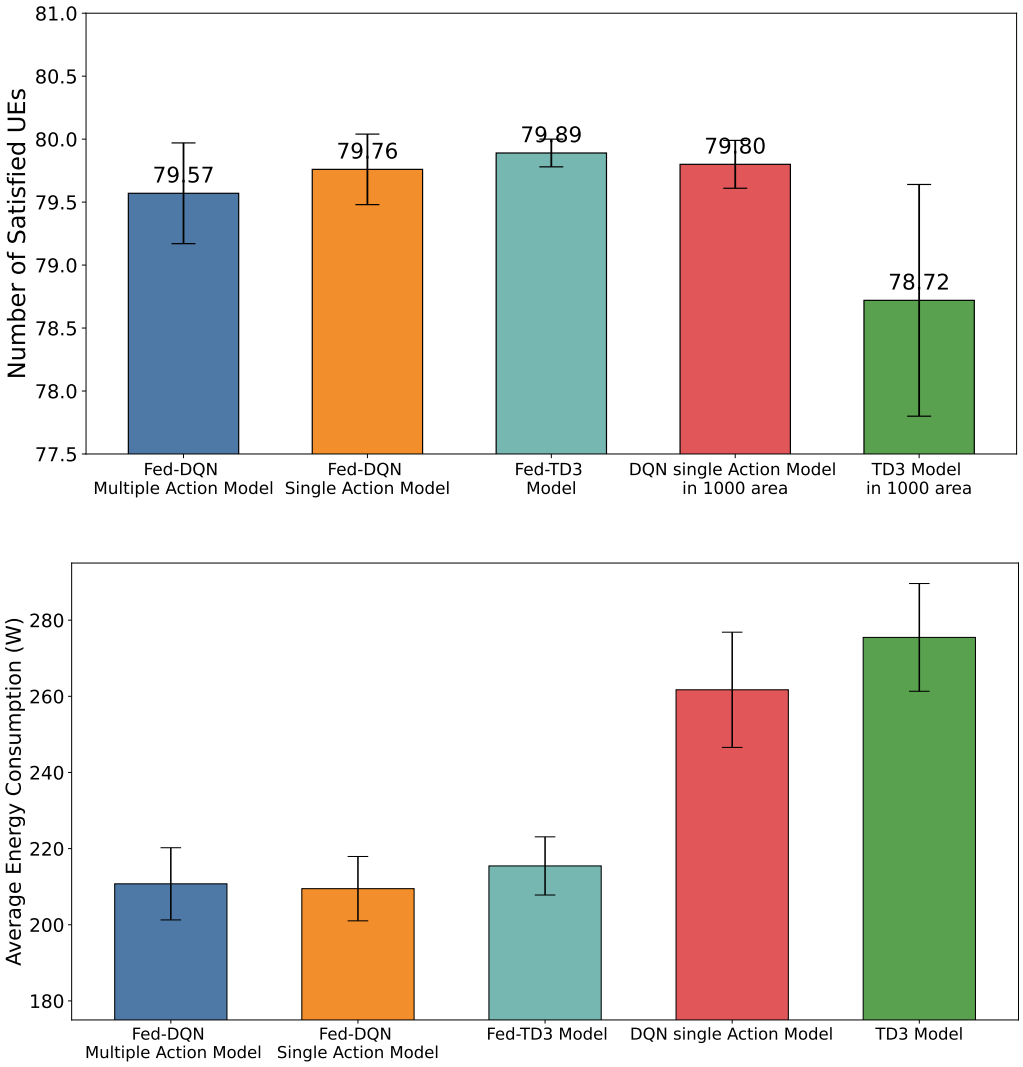


Figure 5.6: Comparison of average energy consumption (right figure) and UE satisfaction (left figure) between federated and centralized reinforcement learning approaches. The first three bars represent federated models (Fed-DQN with multiple and single actions, and Fed-TD3) applied to four independent 500 m×500 m regions with 6 RUs and 20 UEs. The last two bars show the performance of centralized models (DQN and TD3) trained on a merged 1000 m×1000 m region with 24 RUs and 80 UEs.

energy efficiency and user satisfaction. These results confirm that federated reinforcement learning not only enables scalable training across geograph-

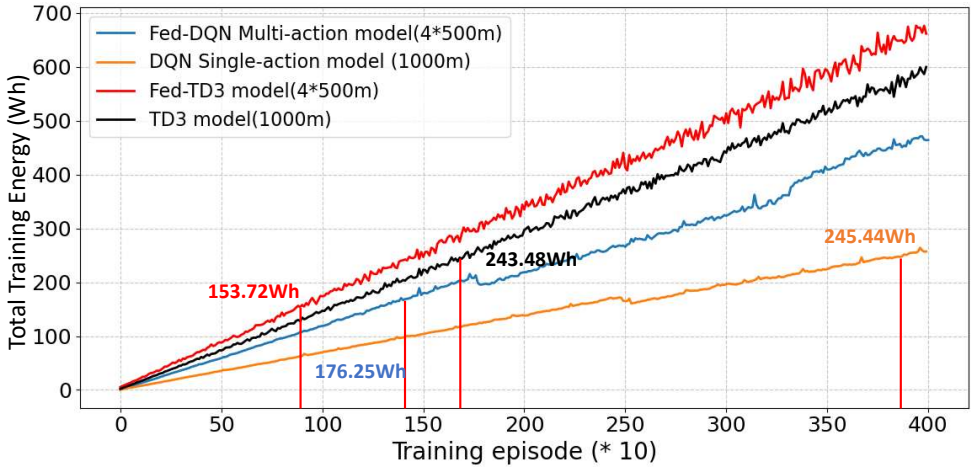


Figure 5.7: The training energy consumption for different DRL models. Each curve represents the cumulative energy usage over time for a specific model. The red labels indicate the total energy consumed by each model upon convergence. The comparison highlights the efficiency differences among federated and centralized approaches.

ically distributed areas, but also yields robust and energy-efficient policies that generalize well when deployed in composite environments. Note that in Fig.5.6, the TD3 model performs slightly worse than the DQNSA in the 1000 m×1000 m area. This is primarily due to the significantly larger action space in the centralized TD3 setting, which involves 24 RUs and poses challenges even with continuous action control. It is important to note that this comparison focuses solely on the differences between FL and non-FL approaches.

It is also important to consider the energy consumed by the AI-based control mechanism itself. In this thesis, the main additional AI-related energy cost comes from offline model training, while online inference in the Near-RT RIC is relatively lightweight compared with the continuous operational power

consumption of the RAN infrastructure. Once trained, the learned policy can be reused for many control intervals, meaning that the training energy cost is amortized over long-term deployment. Therefore, although AI training is not energy-free, its energy cost is expected to be small compared with the cumulative energy savings achieved by switching RUs into sleep mode during network operation. The net benefit of AI-assisted RAN control should therefore be assessed by comparing the one-time or periodic AI training and inference cost with the long-term operational energy saved in the RAN. This trade-off motivates the use of efficient training methods and lightweight inference models in practical O-RAN deployments. Fig. 5.7 reports the total energy consumption required for each model to reach convergence during training. The federated models Fed-TD3 and Fed-DQNMA are compared against centralized DQN and TD3 baselines trained on the full $1000\,\mathrm{m}\times1000\,\mathrm{m}$ area. Each data point represents the accumulated training energy measured at the episode when the model achieves stable reward performance. To obtain accurate measurements, we implemented a custom power monitoring module that continuously samples both CPU and GPU power usage during training. The GPU power was measured using the NVIDIA System Management Interface (nvidia-smi), while CPU utilization was monitored using the `psutil` library. Samples were collected every 100 ms, and total energy consumption (in Wh) was computed by integrating the average power over the training duration. All models were trained exclusively on a single RTX 3060 GPU, with no other significant processes running concurrently. Among all methods, Fed-TD3 exhibits the lowest training energy consumption at 153.72 Wh, followed by Fed-DQNMA at 176.25 Wh. In contrast, the centralized DRL models

consume 243.48 Wh (TD3) and 245.44 Wh (DQN), respectively—the highest among all configurations. Notably, Fed-TD3 reduces training energy by approximately 37.4% compared to the centralized DQN model, while Fed-DQNMA achieves a 28.2% reduction. These results highlight the efficiency advantage of federated reinforcement learning in distributed training scenarios.

5.7 Summary

In this work, we explored energy-efficient RU sleep control within the O-RAN framework through federated DRL strategies. Building on previous insights, we extend the solution to a FL framework aligned with O-RAN's hierarchical control, enabling scalable training across geographically distributed regions without sharing raw data. Extensive simulations across varied network scales and heterogeneous layouts show that the federated global models achieve competitive or superior performance compared to centralized training while significantly improving scalability and generalization. Compared with DQN-based baselines, Fed-TD3 achieves higher rewards, lower training energy consumption, and better adaptability to complex action spaces. These results confirm the practicality of combining centralized DRL optimization with FL to enable flexible and energy efficiency network deployments.

6 Carbon-Aware Reinforcement Learning for Sustainable O-RAN

6.1 Introduction

While the previous chapters have focused on improving the efficiency and scalability of DRL-based RU control in O-RAN systems, an equally critical challenge in next-generation wireless networks is sustainability. As discussed in Chapter 4 and Chapter 5, intelligent RU sleep control can significantly reduce network energy consumption. However, energy efficiency alone does not necessarily translate into environmentally sustainable operation. Recent studies have shown that base stations account for a dominant share of the total energy consumption in radio access networks, contributing significantly to operational carbon emissions [1]. With the increasing integration of renewable energy sources into cellular infrastructure, modern O-RAN deployments are transitioning toward hybrid energy supply models, where some RUs are powered by renewable energy while others rely on conventional grid electricity. In such heterogeneous energy environments, minimizing total energy consumption may not lead to minimal carbon emissions, as different energy sources have fundamentally different carbon footprints.

The work presented in this chapter builds upon the system model and

DRL framework developed in the previous chapters, and extends them to explicitly incorporate carbon awareness into the decision-making process. The underlying system model, including RU deployment, channel modeling, and QoS constraints, remains consistent with Chapter 4. However, the key novelty of this chapter lies in reformulating the RU sleep control problem by integrating carbon emissions into the optimization objective. Specifically, we model the RU control problem as a carbon-aware MDP, where both energy consumption and carbon emissions are explicitly captured in the reward design. We investigate multiple reward formulations that reflect different sustainability objectives, including joint energy–carbon optimization and carbon-oriented optimization. By doing so, we aim to understand how DRL agents adapt their activation strategies under heterogeneous energy supply conditions.

To solve the resulting problem, we employ advanced DRL algorithms, including TD3 and PPO, and evaluate their performance under realistic mobility and 3GPP channel models. Extensive simulations demonstrate that incorporating carbon awareness fundamentally reshapes RU activation patterns, leading to a structural transition from grid-dominant operation to renewable-preferred configurations as renewable penetration increases. This chapter provides a critical step toward sustainable O-RAN design, showing that intelligent control must jointly consider energy efficiency, carbon emissions, and QoS requirements to achieve practical and environmentally responsible network operation.

6.2 Chapter Contributions

The main contributions of this chapter are summarized as follows:

- We extend the conventional energy-centric RU sleep control problem into a sustainability-aware framework by explicitly incorporating heterogeneous energy sources. This reformulation fundamentally changes the structure of the decision-making problem, introducing asymmetric costs for RU activation.

- We develop a carbon-aware MDP formulation in which both energy consumption and carbon emissions are embedded into the reward function. Multiple reward designs are investigated, including joint energy–carbon objectives and carbon-oriented objectives, enabling a systematic analysis of trade-offs between energy efficiency and emission reduction.

- Building upon the DRL frameworks developed in earlier chapters, we evaluate representative deterministic and stochastic policy-learning methods under the proposed carbon-aware formulation. The results reveal that deterministic continuous-control approaches, such as TD3, are more effective in capturing long-term carbon-emission trade-offs in high-dimensional RU activation problems.

- Through extensive simulations under varying renewable penetration levels and spatial deployments, we demonstrate that carbon-aware learning induces a structural shift in RU activation patterns. Specifically, the learned policies progressively prioritize renewable-powered RUs and

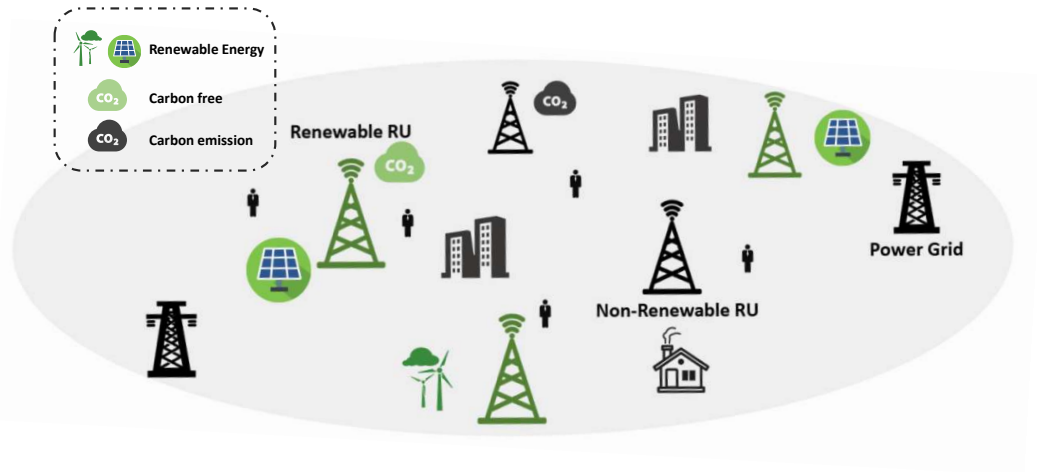


Figure 6.1: Proposed heterogeneous energy network

suppress grid-dependent operation, leading to significant reductions in both energy consumption and carbon emissions.

- This chapter complements the efficiency-focused and scalability-oriented frameworks presented in Chapters 3 and 4 by introducing sustainability as a first-class optimization objective. Together, these contributions establish a unified DRL-based control framework for O-RAN that jointly addresses efficiency, scalability, and environmental impact.

6.3 System Model

We consider an O-RAN based cellular network comprising a fixed set of RUs $\mathcal{M} = \{1, \ldots, M\}$ and a set of UEs $\mathcal{K} = \{1, \ldots, K\}$. The positions of RUs are predetermined and remain unchanged. The operation is time-slotted, with index $t \in \{1, \ldots, T\}$ as presented at Fig. 6.1. To incorporate

sustainability, the RUs are partitioned into two categories: (i) renewable-energy RUs powered by on-site solar generation, denoted by $\mathcal{M}_{\mathrm{R}}$, and (ii) grid-only RUs powered solely by the grid, denoted by $\mathcal{M}_{\mathrm{G}}$, such that

$$\mathcal{M}_{\mathrm{R}} \cup \mathcal{M}_{\mathrm{G}} = \mathcal{M}, \qquad \mathcal{M}_{\mathrm{R}} \cap \mathcal{M}_{\mathrm{G}} = \emptyset.$$

Unlike conventional assumptions where renewable energy is generated locally (e.g., via on-site solar panels), in this work renewable energy is supplied by the power grid. Specifically, we assume that a portion of the grid electricity mix is derived from renewable energy sources (e.g., wind, solar, or hydro), and RUs in $\mathcal{M}_{\mathrm{R}}$ are powered by this renewable energy supply. As such, renewable-powered RUs are not limited by local generation capacity, and sufficient renewable energy is assumed to be available for their operation.

In contrast, RUs in $\mathcal{M}_{\mathrm{G}}$ are powered by non-renewable (fossil-fuel-based) grid electricity, which is associated with non-negligible carbon emissions. Therefore, although both types of RUs are supplied by the grid, they differ in the carbon intensity of their energy sources. This abstraction enables the modeling of heterogeneous energy supply in O-RAN, where the network operator can prioritize low-carbon operation by favoring renewable-powered RUs.

Each RU $m \in \mathcal{M}$ is equipped with Q_m PRBs. Denote by $n_{m,k}^t$ the number of PRBs allocated from RU m to UE k at time slot t. The total PRB usage of RU m is

$$N_m^t = \sum_{k \in \mathcal{K}} n_{m,k}^t, \qquad l_m^t = \frac{N_m^t}{Q_m}, \tag{6.1}$$

where l_m^t is the normalized PRB load.

The SNR of UE k served by RU m at slot t is modeled as

$$\gamma_{m,k}^{t} = \frac{P_{\mathrm{TX}} g_{m,k}^{t}}{Q_m N_0 W_{\mathrm{PRB}}}, \tag{6.2}$$

where P_{TX} is the transmission power, $g_{m,k}^{t}$ is the gain of the channel including path loss and fading, N_0 is the spectral density of the noise power and W_{PRB} is the bandwidth of a PRB. The achievable data rate of UE k is then

$$R_k^t = \sum_{m \in \mathcal{M}} n_{m,k}^{t} \, W_{\mathrm{PRB}} \log_2\!\left(1 + \gamma_{m,k}^{t}\right). \tag{6.3}$$

To guarantee service quality, each UE must satisfy a minimum data rate requirement.

6.3.1 Power Consumption Model

The activation status of RU m is described by a binary variable $\alpha_m^t \in \{0, 1\}$, where $\alpha_m^t = 1$ if RU m is active and $\alpha_m^t = 0$ if RU m is in sleep mode. The power consumption of each RU is composed of three components [9]. The first is fixed power consumption, denoted by $P_m^{\mathrm{Fix},t}$, which represents the energy used by signal processing, cooling systems, and power supply units. This component depends solely on the operational mode of the RU. The second component is the load-dependent power consumption, denoted by $P_m^{\mathrm{data},t}$, primarily attributed to the PA. In this work, the load is quantified based on the PRB utilization, as defined in (6.1). The third component is the transition power $P_m^{\mathrm{trans},t}$, which is incurred only when the RU switches from sleep mode to active mode. By summing these three components, the

total power consumption of RU m at time slot t is given by:

$$P_m^t = P_{\text{fix},m}^t + P_{\text{data},m}^t + P_{\text{trans},m}^t, \tag{6.4}$$

with

$$P_{\text{fix},m}^t = \alpha_m^t P_m^{\text{active}} + (1 - \alpha_m^t) P_m^{\text{sleep}}, \tag{6.5}$$

$$P_{\text{data},m}^t = \alpha_m^t \frac{P_{\text{TX}}}{\eta} l_m^t, \tag{6.6}$$

$$P_{\text{trans},m}^t = |\alpha_m^t - \alpha_m^{t-1}| V_m^{\text{trans}}. \tag{6.7}$$

Here P_m^{active} and P_m^{sleep} are the baseline power levels in active and sleep states, respectively; η is the power amplifier efficiency; and V_m^{trans} captures the overhead due to switching between active and sleep states.

6.3.2 Energy Supply Model

We partition the RUs into two disjoint classes, $\mathcal{M}_{\text{R}}$ and $\mathcal{M}_{\text{G}}$. For each RU m, the instantaneous electrical demand P_m^t is determined by the wireless/sleep model in the previous subsection.

Renewable-energy RUs ($m \in \mathcal{M}_{\text{R}}$) Let $P_{m,\text{ren}}^t \geq 0$ denote the supply contracted from the renewable provider. The power balance is

$$P_m^t \;=\; P_{m,\text{ren}}^t, \tag{6.8}$$

and renewable supply is treated as carbon-neutral in the operational accounting (no on-site storage or curtailment is modeled).

Grid-only sites ($m \in \mathcal{M}_{\mathrm{G}}$) Grid-only RUs have no renewable generation on site. Their demand is fully supplied by the grid:

$$P_m^t \;=\; P_{m,\mathrm{grid}}^t, \qquad P_{m,\mathrm{grid}}^t \geq 0. \tag{6.9}$$

Equations (6.8)–(6.9) together ensure feasibility for both classes while keeping the supply model linear and implementation-agnostic.

6.3.3 Carbon Emission Model

Carbon emissions are attributed solely to the electricity drawn from the utility grid. Let $c_{\mathrm{grid},m}^t$ [kgCO_2/kWh] denote the marginal carbon intensity of the local grid that feeds RU m at slot t; this factor may vary across locations and time. The per-slot emissions of RU m are

$$\mathrm{CO2}_m^t \;=\; \Delta t \, c_{\mathrm{grid},m}^t \, P_{m,\mathrm{grid}}^t, \tag{6.10}$$

where Δt is the slot duration and P_m^t is the total power consumption of RU m. For renewable-powered RUs ($m \in \mathcal{M}_{\mathrm{R}}$), the carbon intensity is assumed to be zero, i.e., $c_{\mathrm{grid},m}^t = 0$, resulting in zero operational carbon emissions. In contrast, for non-renewable-powered RUs ($m \in \mathcal{M}_{\mathrm{G}}$), $c_{\mathrm{grid},m}^t > 0$, and emissions are proportional to their power consumption. This formulation ensures that carbon emissions are directly linked to the type of energy source supplying each RU. The carbon emission model used in this chapter is not proposed as a new carbon accounting model, but is adopted from standard operational carbon accounting principles. In this model, carbon emissions are calculated as the product of grid energy consumption and the carbon

intensity of the electricity supply. Specifically, the carbon emission of grid-powered RU m at time slot t is determined by the amount of grid energy consumed during that time slot and the corresponding grid carbon intensity. This modelling approach is appropriate for the considered O-RAN scenario because the operational carbon footprint of an RU is directly linked to both its power consumption and the energy source used to supply it. Renewable-powered RUs are therefore associated with lower operational emissions, while grid-powered RUs incur emissions according to the carbon intensity of the grid. The model enables a clear evaluation of how carbon-aware RU activation policies can reduce emissions by shifting network operation towards lower-carbon infrastructure where possible.

The network-wide carbon footprint in slot t is the sum over all RUs,

$$\mathrm{CO2}^t \;=\; \sum_{m\in\mathcal{M}} \mathrm{CO2}^t_m \;=\; \Delta t \sum_{m\in\mathcal{M}} c^t_{\mathrm{grid},m}\, P^t_{m,\mathrm{grid}}, \tag{6.11}$$

and the cumulative emissions over a horizon $\{1,\dots,T\}$ read

$$\mathrm{CO2}_{\mathrm{tot}} \;=\; \sum_{t=1}^{T} \mathrm{CO2}^t \;=\; \Delta t \sum_{t=1}^{T} \sum_{m\in\mathcal{M}} c^t_{\mathrm{grid},m}\, P^t_{m,\mathrm{grid}}. \tag{6.12}$$

Renewable utilization is considered carbon-neutral at the operational timescale and hence does not appear in (6.10)–(6.12).

6.3.4 Problem Formulation

Given the system model, the network operation optimizes RU activity $\{\alpha^t_m\}$ over slots $t = 1,\dots,T$. The objective is to minimize a weighted sum of

renewable and non-renewable energy consumption:

$$\min_{\{\alpha_m^t\}} \quad \sum_{t=1}^{T}\left(\lambda \sum_{m\in\mathcal{M}_{\mathrm{R}}} P_m^t + (1-\lambda)\sum_{m\in\mathcal{M}_{\mathrm{G}}} P_m^t\right), \tag{6.13}$$

$$\text{s.t.} \quad R_k^t \;\geq\; R_k^{\min}, \quad \forall k,t, \tag{6.14}$$

$$\sum_{k\in\mathcal{K}} n_{m,k}^t \;\leq\; Q_m, \quad \forall m,t, \tag{6.15}$$

$$\sum_{m\in\mathcal{M}} \delta_{m,k}^t = 1, \quad \forall k,t, \tag{6.16}$$

$$n_{m,k}^t \in \mathbb{Z}_0, \quad \alpha_m^t, \delta_{m,k}^t \in \{0,1\}, \quad \forall m,k,t. \tag{6.17}$$

Here $\mathcal{M}_{\mathrm{R}}$ and $\mathcal{M}_{\mathrm{G}}$ denote the sets of renewable-energy and non-renewable-energy RUs, respectively. P_m^t is the instantaneous power consumption of RU m given its operational state and traffic load. The scalar $\lambda \in [0,1]$ controls the relative emphasis between renewable and non-renewable energy consumption. R_k^t is the downlink rate of UE k at slot t and must satisfy the QoS requirement $R_k^{\min}$. Constraint (6.15) limits the PRB allocation per RU, while (6.16) enforces unique RU association for each UE.

Note that explicit carbon-emission equations are omitted from the formulation. Since only non-renewable consumption contributes to emissions, minimizing the weighted energy cost in (6.13) naturally encourages deactivation of non-renewable RUs, thereby reducing the carbon footprint.

6.4 Sustainable DRL based solution

6.4.1 Markov Decision Process Problem

In this subsection, we formulate the RU sleep/active control problem as a MDP, which enables sequential decision-making under dynamic traffic, user mobility, and heterogeneous energy sources. At each time slot, the agent observes the network state, selects a joint RU activation action, and receives a reward that reflects the sustainability and service performance of the network. The environment then evolves according to user mobility, traffic demand variations, and wireless channel dynamics. This formulation allows the agent to learn long-term policies that balance energy efficiency, carbon emission reduction, and QoS satisfaction over time.

Formally, the MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the state transition probability, and r the reward function. The detailed definitions of the state space, action space, and reward function are given as follows.

State Space

The state comprises essential information required for policy learning. At each time step t, the state includes the real data rates of all UEs, expressed as

$$\mathbf{R}_d^t = \left[R_{d,1}^t, R_{d,2}^t, \ldots, R_{d,K}^t \right]^T, \tag{6.18}$$

which reflects the system's ability to meet user data rate requirements. The activation states of the RUs from the previous time step are given by

$$\boldsymbol{\alpha}^{t-1} = \left[\alpha_1^{t-1}, \alpha_2^{t-1}, \ldots, \alpha_M^{t-1}\right]^T. \tag{6.19}$$

This information captures the temporal coupling between successive RU activation decisions and allows the agent to account for switching behavior.

The PRB utilization levels of the RUs are represented as

$$\mathbf{L}^t = \left[l_1^t, l_2^t, \ldots, l_M^t\right]^T, \tag{6.20}$$

which quantify the instantaneous traffic load of each RU. In addition, the spatial positions of the UEs are included as

$$\mathbf{U}^t = \left[(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_K^t, y_K^t)\right]^T, \tag{6.21}$$

where (x_k^t, y_k^t) denote the coordinates of UE k at time t. The UE positions directly affect path loss, user association, and achievable data rates, and thus influence RU load distributions. Including spatial information in the state enables the agent to anticipate future handovers, traffic shifts, and coverage-critical regions, rather than reacting solely to instantaneous load conditions.

Overall, the state at time t is defined as

$$s_t = \left[\mathbf{R}_d^t, \boldsymbol{\alpha}^{t-1}, \mathbf{L}^t, \mathbf{U}^t\right]^T. \tag{6.22}$$

All state variables are normalized to comparable ranges to facilitate stable and efficient learning.

Action Space

The action $a_t \in \mathcal{A}$ represents the operational states of all RUs and is defined as

$$\mathcal{A} = [\alpha_1, \alpha_2, \ldots, \alpha_M], \tag{6.23}$$

where each element $\alpha_m \in \{0, 1\}$ denotes the operating mode of RU m. Specifically, $\alpha_m = 1$ indicates that RU m is active, while $\alpha_m = 0$ corresponds to sleep mode. The action is generated by the DRL policy and determines the network-wide RU activation pattern at each time slot.

Reward Function

The reward function is designed to guide the agent toward sustainability-aware RU control decisions by balancing energy efficiency, carbon emission reduction, and QoS satisfaction. To evaluate different sustainability objectives, we consider two carbon-aware reward formulations.

a) Energy–Carbon–QoS reward:

$$r_t^{(1)} = -w_1 \cdot \mathrm{norm}\big(P_{\mathrm{tot}}^t\big) - w_2 \cdot \mathrm{norm}\big(C_{\mathrm{tot}}^t\big) - w_3 \cdot \mathrm{norm}\big(K_{\mathrm{unsat}}^t\big), \tag{6.24}$$

where P_{tot}^t denotes the total network energy consumption at time t, C_{tot}^t represents the corresponding carbon emissions, and K_{unsat}^t is the number of UEs whose data rates fall below the minimum QoS requirement. Each term is normalized to ensure comparable scales and to prevent dominance by any single component. This reward formulation jointly optimizes energy efficiency, carbon reduction, and service reliability.

b) Carbon–QoS-oriented reward:

$$r_t^{(2)} = -w_4 \cdot \mathrm{norm}\big(C_{\mathrm{tot}}^t\big) - w_5 \cdot \mathrm{norm}\big(K_{\mathrm{unsat}}^t\big)\,, \tag{6.25}$$

which focuses explicitly on minimizing operational carbon emissions while maintaining QoS constraints. By removing the explicit energy term, this reward highlights scenarios where carbon reduction, rather than energy minimization, is the dominant objective, particularly in networks with heterogeneous renewable-powered and grid-powered RUs.

In both reward formulations, QoS constraints $R_{d,k}^t \geq R_{\min,d,k}$ are enforced through the penalty term associated with K_{unsat}^t. The temporal accumulation of this penalty discourages persistent QoS violations and promotes stable network operation. At each time step, the agent observes the current state s_t, selects an action a_t according to its policy $\pi(a_t|s_t)$, and receives the corresponding reward. The policy is then updated to maximize the expected long-term cumulative reward, enabling the agent to learn optimal sustainability-aware RU sleep/active control strategies.

6.4.2 PPO-Based Training for RU Sleep Control

To provide a representative stochastic policy-gradient baseline, we implement PPO for the same RU sleep/active control task and environment as the TD3 agent. In our setting, the controller must decide, at every time slot, whether each RU should be active or sleeping, resulting in a multi-dimensional binary action vector $\mathbf{a}_t \in \{0,1\}^M$. Unlike TD3, which learns a deterministic continuous policy and applies thresholding for execution, PPO directly models the joint RU activation decision as a stochastic policy over binary actions. This

design enables on-policy learning with clipped policy updates while preserving the discrete nature of RU switching.

Policy Parameterization for Multi-RU Binary Actions

We adopt an actor-critic architecture with a shared feature extractor and two heads: an actor head producing RU-wise logits and a critic head estimating the state value. Given the observed state s_t, the network outputs

$$\boldsymbol{\ell}_t,\, V_\psi(s_t) = f_\psi(s_t), \tag{6.26}$$

where $\boldsymbol{\ell}_t \in \mathbb{R}^M$ denotes the logits for all RUs, and $V_\psi(s_t)$ is the state-value estimate. The RU activation policy is modeled as a factorized Bernoulli distribution

$$\pi_\psi(\mathbf{a}_t|s_t) = \prod_{m=1}^{M} \text{Bernoulli}(a_{t,m};\, p_{t,m})\,, \quad p_{t,m} = \sigma(\ell_{t,m}/\tau_t)\,, \tag{6.27}$$

where $\sigma(\cdot)$ is the sigmoid function and τ_t is a temperature parameter. Sampling from this distribution yields the RU activation vector $\mathbf{a}_t$, which is executed directly in the environment. This formulation maps naturally to RU sleep scheduling: each dimension corresponds to turning one RU active/sleep, and the stochastic policy allows PPO to explore different activation patterns under the same state.

Exploration Design in RU Scheduling

To avoid premature saturation of Bernoulli probabilities (i.e., $p_{t,m} \to 0$ or 1 leading to near-zero entropy and poor exploration), we incorporate several

exploration mechanisms during training. First, we inject Gaussian noise into the actor logits before sampling,

$$\tilde{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \sigma_{\ell,t}^2 \mathbf{I}), \tag{6.28}$$

and apply temperature scaling through τ_t to control stochasticity. Both $\sigma_{\ell,t}$ and τ_t are annealed over training to transition from exploration to exploitation. Second, an entropy bonus is added to the objective to encourage diverse RU activation decisions at early stages. These mechanisms are particularly important in our scenario because RU control is high-dimensional and strongly coupled via traffic load, user association, and QoS constraints; insufficient exploration may cause the policy to collapse to a fixed RU pattern that fails to adapt to changing UE distributions.

Rollout Collection and Advantage Estimation with Episode Boundaries

PPO is trained on on-policy rollouts collected by interacting with the same environment dynamics used by TD3, including mobility, handovers, and the RU-specific energy/carbon models. At each time step, the transition $(s_t, \mathbf{a}_t, r_t, s_{t+1})$ is recorded along with the log-probability $\log \pi_\psi(\mathbf{a}_t|s_t)$ and value estimate $V_\psi(s_t)$. We collect rollouts for a fixed number of steps and then perform multiple epochs of PPO updates.

To compute low-variance policy gradients, we use generalized advantage estimation (GAE). Importantly, in our implementation, advantage accumulation is reset at episode boundaries (either environment termination or time-limit truncation), preventing leakage of advantages across independent

episodes:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\psi(s_{t+1}) - V_\psi(s_t). \tag{6.29}$$

This is crucial for RU scheduling since episodes are time-limited and the environment may reset due to simulation horizon; improper bootstrapping would distort the estimated returns and destabilize training.

Clipped PPO Objective for RU Sleep Decisions

Given a batch of rollout data, PPO updates the policy by maximizing the clipped surrogate objective:

$$\mathcal{L}_{\text{PPO}}(\psi) = \mathbb{E}\left[\min\left(\rho_t(\psi)\hat{A}_t, \text{clip}(\rho_t(\psi), 1-\epsilon, 1+\epsilon)\,\hat{A}_t\right)\right], \tag{6.30}$$

where

$$\rho_t(\psi) = \exp\left(\log \pi_\psi(\mathbf{a}_t|s_t) - \log \pi_{\psi_{\text{old}}}(\mathbf{a}_t|s_t)\right), \tag{6.31}$$

and ϵ is the clipping parameter. The value function is trained with a squared error loss, and an entropy regularizer is used to promote exploration:

$$\mathcal{L}(\psi) = -\mathcal{L}_{\text{PPO}}(\psi) + c_v\mathbb{E}\left[(\hat{R}_t - V_\psi(s_t))^2\right] - c_H\mathbb{E}[\mathcal{H}(\pi_\psi(\cdot|s_t))], \tag{6.32}$$

where $\hat{R}_t$ is the estimated return, $\mathcal{H}(\cdot)$ denotes entropy, and c_v and c_H control the value and entropy terms, respectively. In the RU sleep control context, this objective updates the RU activation probabilities so that actions yielding lower energy/carbon cost under QoS constraints become more likely, while the clipped update prevents abrupt policy shifts that could cause unstable

RU switching.

Stability Measures and Implementation Details

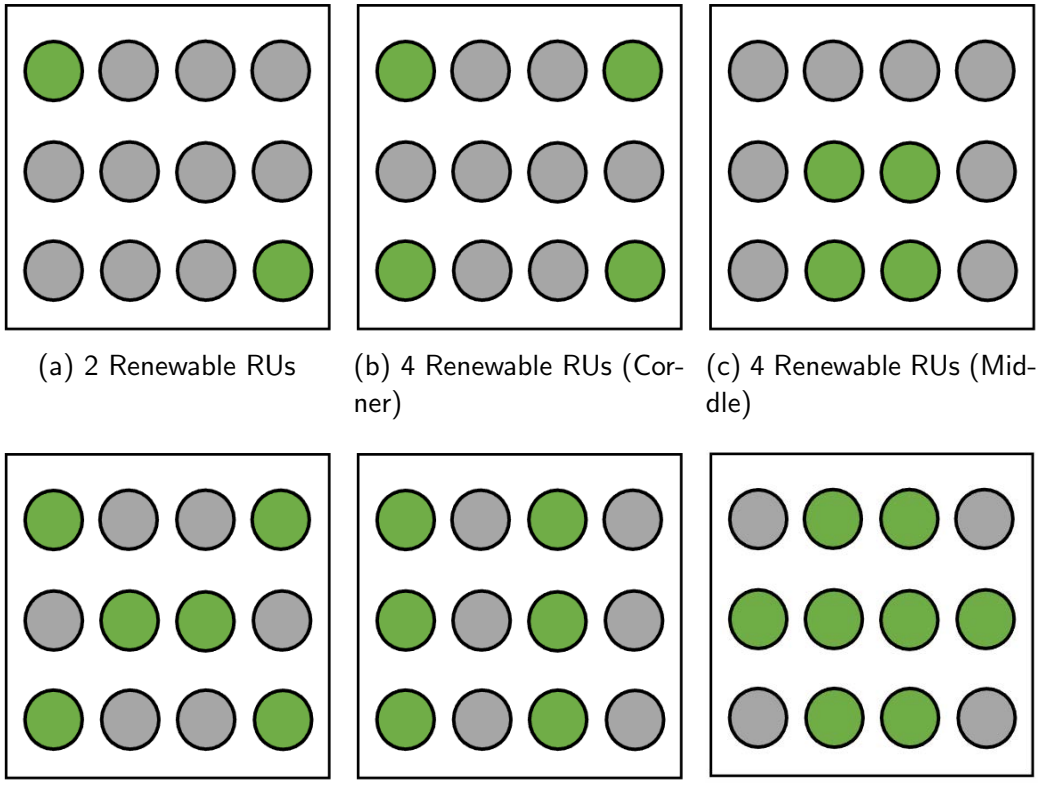
To improve numerical stability in high-dimensional binary control, we clamp logits to bounded ranges before computing log-probabilities and apply gradient norm clipping during optimization. Moreover, reward normalization is applied per rollout to reduce scale sensitivity and to stabilize the advantage estimates. The actor and critic are optimized jointly using separate learning rates for the shared backbone/actor head and the critic head. These design choices help mitigate training collapse (e.g., entropy going to zero) that can occur when the policy quickly becomes overconfident in a particular RU activation pattern.

Overall, PPO observes the same state representation as TD3 and outputs a stochastic RU activation vector $\mathbf{a}_t$ that is executed directly in the environment. This provides a strong and widely adopted policy-gradient baseline for evaluating the suitability of stochastic versus deterministic control paradigms in sustainability-aware RU sleep scheduling.

6.5 Simulation Environment

6.5.1 Simulation Settings

In our simulations, each scenario is modeled as an O-RAN environment where M RUs serve K UEs within a square area of size $L \times L\,\mathrm{m}^2$. The considered scenarios explicitly differentiate between renewable-powered RUs and grid-powered RUs, enabling the evaluation of sustainability-aware RU sleep



(a) 2 Renewable RUs (b) 4 Renewable RUs (Corner) (c) 4 Renewable RUs (Middle)

(d) 6 Renewable RUs (Random) (e) 6 Renewable RUs (Even) (f) 8 Renewable RUs

Figure 6.2: Spatial deployment scenarios with different renewable RU penetration levels and layouts. Green circles indicate renewable-powered RUs, while gray circles represent grid-powered RUs. These scenarios are used to evaluate the robustness of RU sleep scheduling policies under varying renewable availability and spatial distributions.

control under heterogeneous energy sources. Renewable-powered RUs are assumed to operate with zero operational carbon emissions, while grid-powered RUs incur both energy consumption and carbon emissions according to the adopted energy and carbon models.

In practical O-RAN deployments, the Near-RT RIC is typically hosted at edge cloud or regional edge data centers, with latency budgets ranging from 10 ms to 1 s. In our system model, the inference of the DRL agents is

Table 6.1: Simulation Parameters

Parameter	Value
DRL Training Parameters	
TD3 actor learning rate (α_π)	1×10^{-4}
TD3 critic learning rate (α_Q)	1×10^{-3}
PPO learning rate (α_{PPO})	3×10^{-4}
PPO clipping parameter (ϵ)	0.2
GAE parameter (λ)	0.95
Entropy coefficient (c_H)	0.01
Value loss coefficient (c_V)	0.5
Mini-batch size (B)	128
Replay buffer size (TD3 only)	5×10^4
Discount factor (γ)	0.99
Training episodes (N_{eps})	2000
Soft update coefficient (TD3, τ)	0.01
Random seed	42
Mobility and Traffic Parameters	
Average UE speed (v_{avg})	2 m/s
Std. deviation of UE speed (v_{std})	0.5 m/s
Minimum data rate ($R_{\min}$)	6 Mbps
Energy and Carbon Model Parameters	
Active mode RU power (P^{active})	20 W
Sleep mode RU power (P^{sleep})	5 W
Mode transition power (P^{trans})	3 W
Maximum transmit power (P_{TX})	1 W (30 dBm)
Power amplifier efficiency (η)	0.5
Reward weights (w_1, w_2, w_3)	$(1, 3, 10)$
Reward weights (w_4, w_5)	$(1, 2.5)$

executed within the Near-RT RIC as an xApp, ensuring that RU sleep/active decisions satisfy the real-time control requirements. Model training is performed offline using simulated interaction data, while the trained policies are deployed for online inference. This deployment setting aligns with current O-RAN architectural principles and ensures realistic decision-making latency.

As illustrated in Fig. 6.2, multiple spatial layouts are considered to examine the impact of renewable penetration and spatial distribution. The

Table 6.2: Comparison of Reward Function Designs

Reward function	Description
AG1	Energy + Carbon + QoS
AG2	Carbon + QoS
AG3	Energy + QoS

scenarios include different numbers of renewable-powered RUs (e.g., 2, 4, 6, and 8), as well as different spatial arrangements such as centralized, evenly distributed, and random placements. These layouts are designed to evaluate the robustness of the learned policies under varying renewable availability and spatial heterogeneity.

UEs move according to a periodic mobility model with constant speed $v = v_{\text{avg}} \pm v_{\text{std}}$, where the mean speed is $v_{\text{avg}} = 2\,\text{m/s}$ and the standard deviation is $v_{\text{std}} = 0.5\,\text{m/s}$. Individual UE speeds are randomly drawn from this range. UEs periodically travel from the boundary of the service area toward the center and back, forming a structured cyclic mobility pattern. This mobility model induces time-varying traffic loads and handovers across RUs, which is critical for evaluating adaptive RU sleep control policies. The wireless channel model follows the 3GPP UMi specification [77] with both LOS and NLOS conditions.

Each training episode consists of 200 time steps. To account for RU state transition latency and to match practical network operation timescales, each time step corresponds to 1.75 s. RU sleep/active decisions, energy consumption measurement, and carbon emission accounting are all performed at this temporal granularity. The instantaneous energy and carbon emission values are recorded at each step, while episode-level metrics are obtained by accumulation over the entire episode.

We evaluate two representative DRL approaches for centralized RU sleep control: the deterministic policy-based TD3 and the stochastic policy-gradient-based PPO. Both agents share the same state representation, action space, and environment dynamics to ensure a fair and consistent comparison. To investigate the impact of sustainability-aware optimization, three different reward formulations are considered, are summarized in Table 6.2. AG1 jointly accounts for normalized energy consumption, carbon emissions, and QoS satisfaction, enabling a balanced trade-off between efficiency and sustainability. AG2 focuses on carbon-aware optimization by considering carbon emissions and QoS satisfaction only, while AG3 represents a conventional energy-driven design that optimizes energy consumption and QoS without explicitly incorporating carbon emissions. This setup allows us to systematically analyze the interaction and trade-offs between energy efficiency and carbon reduction. Detailed network architectures, optimizer configurations and system parameters are summarized in Table 6.1.

To provide a non-learning baseline, a green-aware hysteresis heuristic is implemented. The heuristic makes RU activation decisions directly from the current network state without training a neural network. At each time step, the policy first computes the number of UEs associated with each RU, the PRB utilisation of each RU, and the average user satisfaction. The PRB utilisation of RU m is calculated as

$$U_m = \frac{N_{\mathrm{RB},m}^{\mathrm{used}}}{N_{\mathrm{RB}}},$$

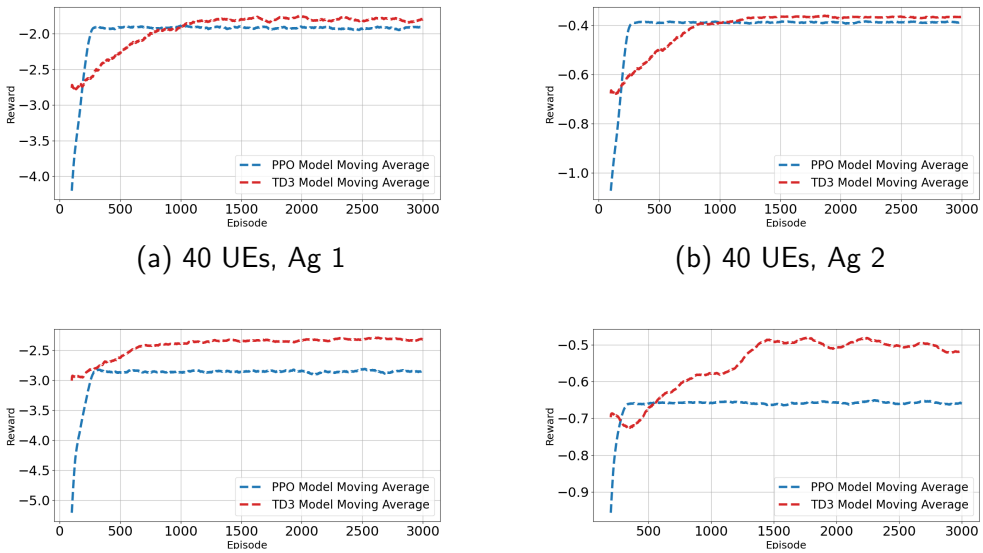
where $N_{\mathrm{RB},m}^{\mathrm{used}}$ is the number of allocated resource blocks at RU m, and $N_{\mathrm{RB}} = 277$ is the total number of PRBs per RU.

Table 6.3: Heuristic baseline parameters.

Parameter	Value
Idle PRB utilisation threshold, U_{off}	0.20
Congestion PRB utilisation threshold, U_{on}	0.90
Minimum average user satisfaction, $S_{\min}$	0.95
Idle hysteresis duration, T_{off}	0 time steps
Maximum additionally activated RUs per step	2
Total PRBs per RU, N_{RB}	277

The heuristic follows five rules. First, any RU serving at least one UE is kept active. Second, an RU is considered idle if it has no associated UEs and its PRB utilisation is below the idle threshold U_{off}. Third, an idle RU is switched off after it has remained idle for T_{off} consecutive time steps. Fourth, if the average user satisfaction falls below $S_{\min}$, or if any active RU has PRB utilisation above the congestion threshold U_{on}, the heuristic activates nearby sleeping RUs to relieve congestion. When multiple candidate RUs are available, renewable-powered RUs are prioritised before grid-powered RUs. Finally, at least one RU is always kept active as a safety condition.

The parameter values used for the heuristic baseline are summarised in Table 6.3. The idle threshold is set to $U_{\text{off}} = 0.20$, the congestion threshold is set to $U_{\text{on}} = 0.90$, and the minimum average user satisfaction is set to $S_{\min} = 0.95$. The hysteresis duration is $T_{\text{off}} = 0$, meaning that an idle RU can be switched off immediately once the idle condition is satisfied. To avoid activating too many RUs in one control step, the maximum number of additionally activated RUs is limited to two. The heuristic is evaluated over 50 episodes, with 200 time steps per episode.



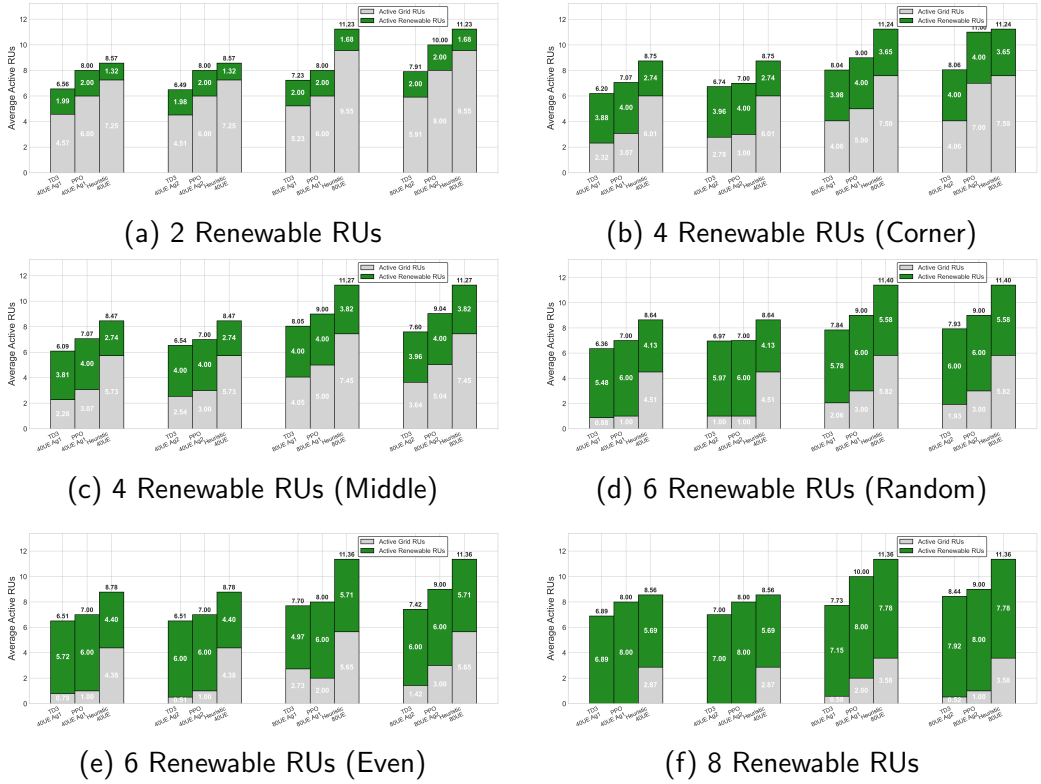
(a) 40 UEs, Ag 1 (b) 40 UEs, Ag 2

(c) 80 UEs, Ag 1 (d) 80 UEs, Ag 2

Figure 6.3: Training convergence comparison between PPO and TD3 under the 6 renewable RU (random layout) scenario. The moving-average episode rewards are shown for 40 UE and 80 UE traffic loads, each evaluated with two independent reward functions (AG1 and AG2).

6.5.2 Numerical Results

Fig. 6.3 presents the training convergence behavior of TD3 and PPO under the 6-renewable-RU random deployment scenario for both 40 UEs and 80 UEs traffic loads. Two reward formulations are considered: AG1, which jointly accounts for energy consumption, carbon emissions, and QoS satisfaction, and AG2, which emphasizes carbon emissions and QoS without explicitly penalizing energy consumption. For the moderate traffic load (40 UEs), both algorithms converge within approximately the first 1000 episodes. However, distinct convergence characteristics can be observed. PPO exhibits a faster initial improvement during the early training phase, rapidly escaping low-reward regions due to its stochastic exploration mechanism. In contrast,



(a) 2 Renewable RUs

(b) 4 Renewable RUs (Corner)

(c) 4 Renewable RUs (Middle)

(d) 6 Renewable RUs (Random)

(e) 6 Renewable RUs (Even)

(f) 8 Renewable RUs

Figure 6.4: Average number of active renewable-powered and grid-powered RUs under different control strategies for various renewable penetration levels and spatial layouts. Each subfigure corresponds to a different deployment scenario. Results are shown for both 40 UEs and 80 UEs.

TD3 improves more gradually but achieves consistently higher final reward values under both AG1 and AG2. This suggests that the deterministic policy structure of TD3 is more effective in refining high-dimensional binary RU activation decisions once a near-optimal region is identified. Under the heavier traffic load (80 UEs), the performance gap between the two algorithms becomes more pronounced. The reward landscape in this case is more complex due to increased QoS constraints and tighter resource coupling among RUs. PPO converges quickly to a stable region but remains trapped in a comparatively suboptimal solution. TD3, on the other hand, continues to im-

prove over a longer training horizon and achieves higher asymptotic rewards. This behavior indicates that deterministic continuous-control methods are better suited for capturing the structured trade-offs in RU sleep scheduling under high load conditions. Comparing AG1 and AG2 reveals additional insights. When both energy and carbon terms are included (AG1), convergence is slightly smoother and more stable, as the energy term regularizes excessive RU activation. In AG2, where only carbon emissions are penalized, the reward landscape becomes sharper due to asymmetric penalties between renewable and grid-powered RUs. This results in increased learning difficulty, particularly for PPO, whose stochastic updates may struggle in highly asymmetric reward environments. TD3 remains more stable across both formulations, suggesting stronger robustness to reward shaping variations. Overall, the results demonstrate that while PPO provides competitive early-stage exploration, TD3 achieves superior long-term convergence and stability in sustainability-aware RU sleep control.

Fig. 6.4 presents the average number of active RU powered by renewable energy and RU powered by the grid under different scenarios of renewable deployment, evaluated under both 40 UEs and 80 UEs traffic conditions. A clear structural transition can be observed as renewable penetration increases. In low-renewable scenarios (e.g., 2 renewable RUs), all learning-based strategies activate nearly all available renewable RUs while still relying substantially on grid-powered RUs to satisfy coverage and rate constraints. As the number of renewable RUs increases, both TD3 and PPO progressively reduce grid-powered activation and shift the operational structure toward renewable-dominant configurations. In the 8-renewable-RU scenario

under 40 UEs, the TD3-based policy nearly eliminates grid-powered activation, demonstrating strong renewable prioritization induced by the carbon-oriented reward formulation. Across all renewable layouts and traffic loads, TD3 consistently activates average 1 fewer RUs than PPO and 2.5 fewer RUs than heuristic model . This difference becomes more pronounced under higher traffic demand (80 UEs), where PPO maintains a higher reliance on grid-powered infrastructure to stabilize QoS performance. The results indicate that TD3 more effectively captures long-term carbon-emission trade-offs under discrete RU switching constraints, enabling a more aggressive yet stable migration toward renewable-powered operation. The impact of reward design is also evident. Under AG2 (carbon-only reward), both TD3 and PPO exhibit stronger renewable prioritization compared to AG1, which jointly considers energy consumption and carbon emissions. AG2 drives policies that maximize renewable utilization even at the expense of slightly higher total active RUs, whereas AG1 enforces a more balanced trade-off between energy efficiency and emission reduction. This confirms that the reward formulation directly reshapes the activation structure of the network. Compared to the heuristic baseline, the DRL-based strategies achieve a clear structural advantage. The heuristic policy activates a larger number of grid-powered RUs and shows limited sensitivity to renewable spatial distribution. In contrast, TD3 and PPO adapt to heterogeneous renewable layouts and dynamically reconfigure RU activation to exploit renewable availability. Overall, the results demonstrate that increasing renewable penetration progressively transforms the network from a grid-dominant to a renewable-dominant operational regime, with TD3 achieving the most effective carbon-aware structural

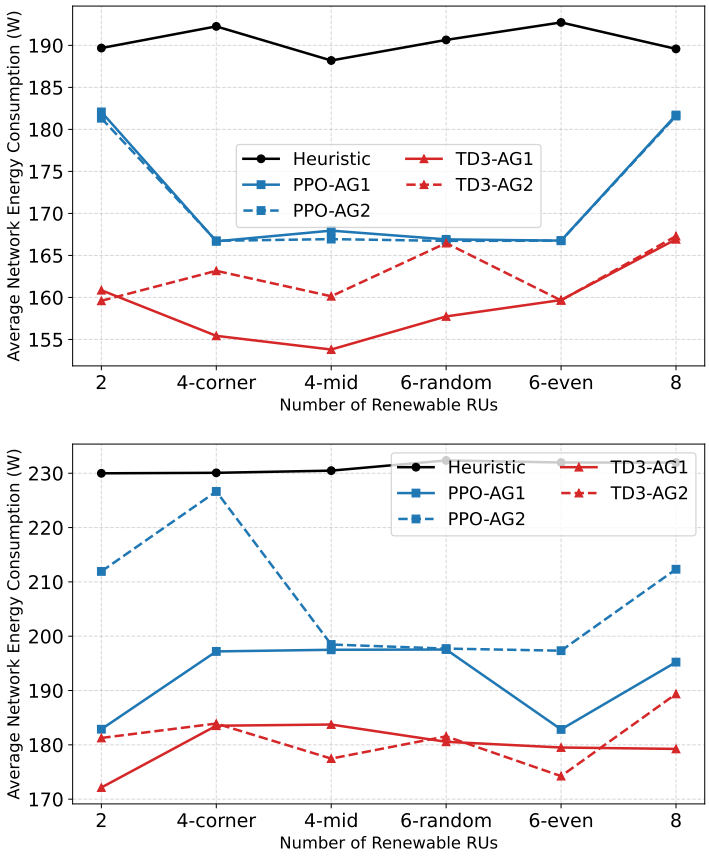


Figure 6.5: Average network energy consumption under different control strategies as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. Results are shown for TD3, PPO with two reward functions (AG1 and AG2) and heuristic baselines.

reconfiguration across all evaluated scenarios.

Figure 6.5 illustrates the average network energy consumption as a function of renewable RU penetration for both 40 UE and 80 UE scenarios. For the 40 UE case, the heuristic baseline exhibits the highest energy consumption across all renewable configurations, remaining above 180 W in low-renewable scenarios and only gradually decreasing as renewable penetration increases. In contrast, all DRL-based approaches achieve noticeable energy savings. Specifically, TD3-based strategies consistently outperform

PPO under both reward formulations. For example, in the 2-renewable scenario, TD3 reduces energy consumption by approximately 5–10 W compared to PPO, corresponding to roughly a 3–6% relative improvement. This gap persists across intermediate deployments (4 and 6 renewable RUs) and remains visible even in the 8-renewable configuration, demonstrating that TD3 learns a more energy-efficient RU activation structure. A similar pattern is observed in the 80 UE scenario, although absolute consumption levels are higher due to increased traffic demand. Energy consumption ranges roughly between 220–230 W under low renewable penetration and decreases toward 190–200 W as renewable RUs increase. TD3 again achieves the lowest energy consumption in most configurations, with typical savings of 5–8 W compared to PPO, confirming its superior capability in handling large discrete activation spaces under heavier load conditions. The impact of reward design is also evident. Across most renewable deployments, AG1 achieves lower energy consumption than AG2. While AG2 strongly prioritizes renewable utilization, it occasionally activates additional RUs to minimize carbon emissions, leading to slightly higher total energy usage. In contrast, AG1 explicitly penalizes normalized energy consumption, resulting in more compact activation patterns and lower overall power draw. This difference becomes more pronounced in moderate renewable scenarios (4 and 6 RUs), where AG1 typically provides an additional 2–4 W energy reduction compared to AG2.

To further investigate the impact of carbon-aware optimization on energy efficiency, Fig. 6.6 introduces an additional reward formulation (AG3) that considers only average network energy consumption and QoS constraints, without including the carbon-emission term. As shown in Fig. 6.6, AG3

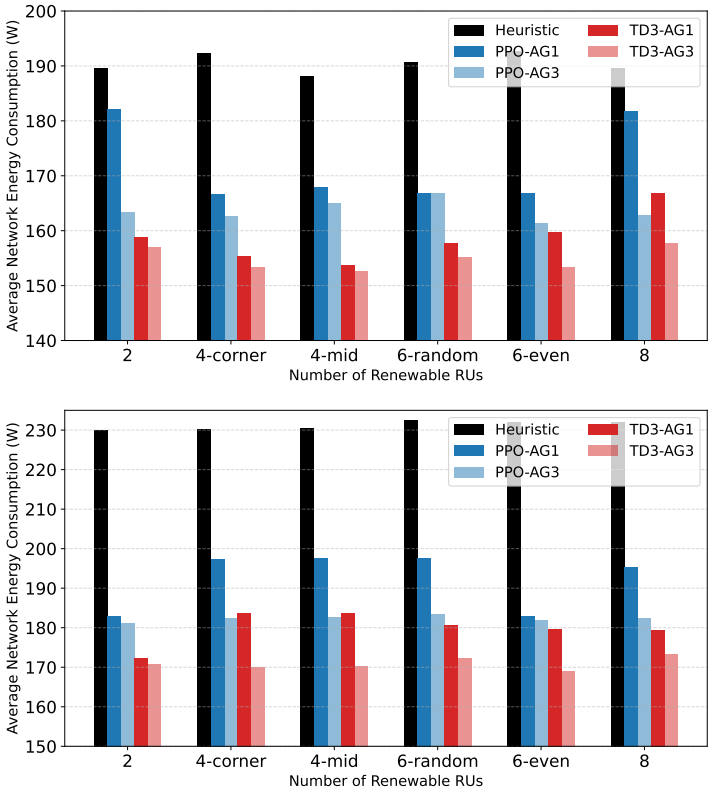


Figure 6.6: Average network energy consumption under different control strategies as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. Results are shown for TD3, PPO with two reward functions (AG1 and AG3) and heuristic baselines.

consistently achieves lower network energy consumption than AG1 in most renewable deployment scenarios for both traffic loads. This behavior is expected since AG3 focuses solely on minimizing energy consumption, whereas AG1 also accounts for carbon emissions. In carbon-aware settings, the agent may activate additional renewable-powered RUs to reduce carbon emissions even if this slightly increases the total network energy consumption. These results highlight the inherent trade-off between pure energy minimization and carbon-aware network operation. While AG3 provides the lowest energy consumption, AG1 achieves a more balanced solution by jointly optimizing

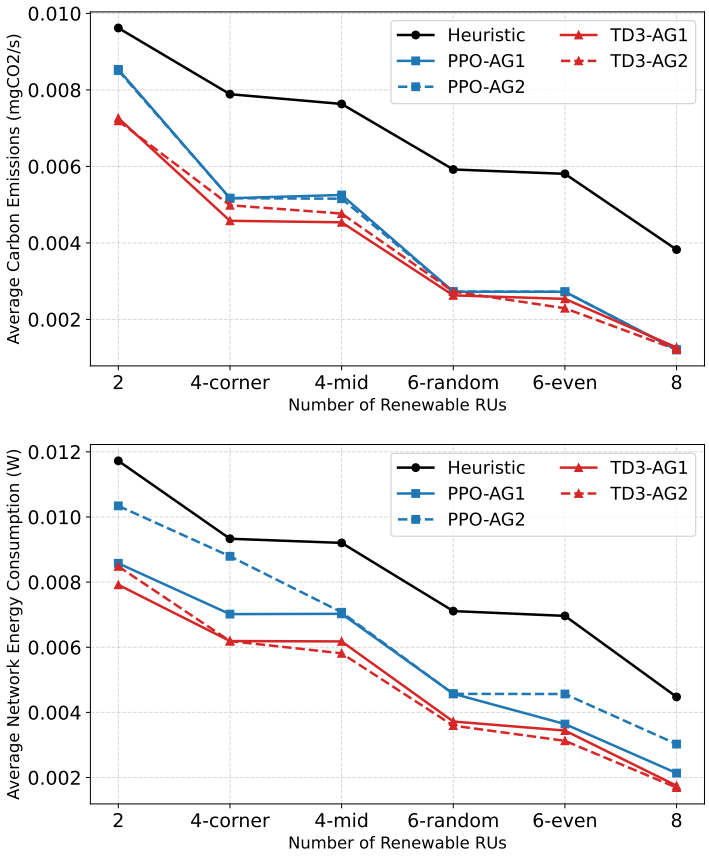


Figure 6.7: Average operational carbon emissions under different control strategies as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs. The results illustrating how carbon-aware control policies increasingly rely on renewable-powered RUs and reduce grid-dependent emissions as renewable availability grows.

energy efficiency and carbon reduction.

Figure 6.7 presents the average operational carbon emissions under different renewable penetration levels for both 40 UE and 80 UE scenarios. A clear monotonic reduction in carbon emissions is observed as the number of renewable RUs increases from 2 to 8. This confirms that increasing renewable deployment structurally shifts the network from grid-dependent operation toward renewable-dominant activation patterns. For the 40 UE scenario (top

panel), TD3 consistently achieves the lowest carbon emissions across all renewable configurations. In low-renewable scenarios (e.g., 2 renewable RUs), TD3-AG2 reduces carbon emissions by approximately 8–12% compared to PPO-AG2 and by more than 15% compared to the heuristic baseline. As renewable penetration increases to 6 and 8 RUs, the relative improvement remains stable, with TD3 maintaining a 6–10% reduction over PPO and over 18% compared to heuristic control. This indicates that TD3 more effectively suppresses unnecessary grid-powered RU activation, especially under constrained renewable availability. In the 80 UE scenario (bottom panel), overall carbon emissions increase due to higher traffic demand; however, the relative performance ordering remains consistent. TD3-based policies again achieve the lowest emissions across all renewable levels. Under moderate renewable penetration (4–6 RUs), TD3-AG2 achieves approximately 10% lower emissions than PPO-AG2 and up to 20% lower emissions than the heuristic strategy. The performance gap becomes more pronounced under high load conditions, demonstrating that TD3 is more capable of learning stable long-term carbon-minimizing activation patterns when the system operates closer to capacity. The impact of reward formulation is also evident. AG2 (carbon-only optimization) consistently yields lower emissions than AG1 for both TD3 and PPO, confirming that explicitly prioritizing carbon in the reward function drives stronger renewable utilization. However, AG1 achieves competitive emission reduction while maintaining energy-awareness, as discussed in Fig. 5. This highlights the inherent trade-off between pure carbon minimization and joint energy–carbon optimization.

Figure 6.8 further compares the average network carbon emissions under

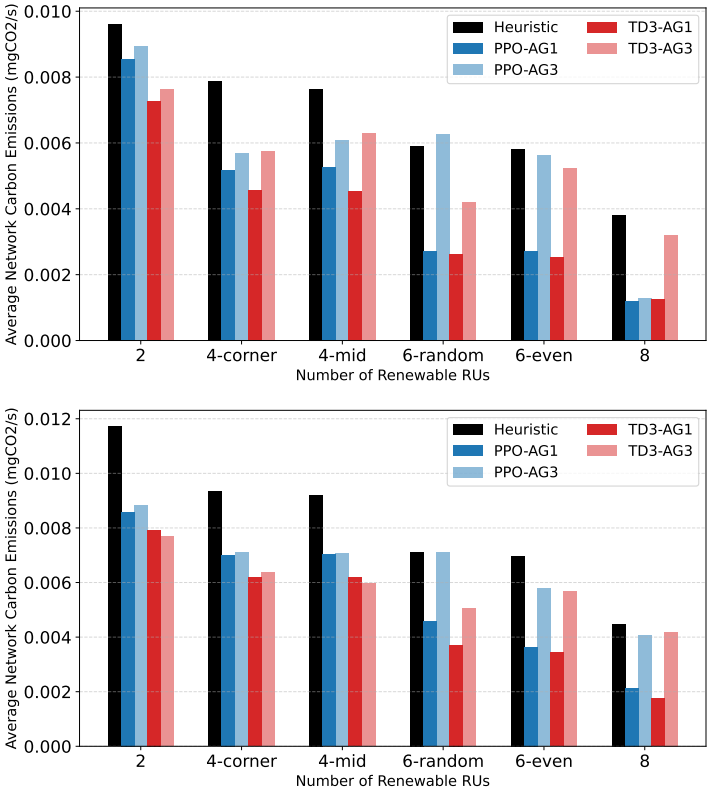


Figure 6.8: Average operational carbon emissions under different control strategies (AG1 and AG3) as a function of the number of renewable RUs for the scenario with 40 (top) and 80 (bottom) UEs.

AG1 and AG3 reward formulations for both 40 UE and 80 UE scenarios. As observed in Fig. 6.8, AG1 consistently achieves lower carbon emissions than AG3 across all renewable deployment configurations. Quantitatively, AG1 reduces carbon emissions by approximately 12%–28% compared to AG3, with an average reduction of around 18% across different scenarios. This improvement can be attributed to the explicit inclusion of carbon emissions in the AG1 reward formulation, which encourages the agent to prioritize renewable-powered RUs and minimize reliance on grid-powered infrastructure. In contrast, AG3 optimizes only network energy consumption and QoS constraints

without accounting for the carbon intensity of grid energy. Consequently, although AG3 often achieves lower total energy consumption, it tends to activate more grid-powered RUs, leading to higher carbon emissions. These results Fig. 6.8 and Fig. 6.6 clearly demonstrate the trade-off between energy minimization and carbon-aware optimization. While AG3 favors purely energy-efficient operation, AG1 enables a more sustainable network behavior by explicitly guiding the policy toward low-emission configurations, achieving significant carbon reduction at the cost of a marginal increase in energy consumption.

6.6 Conclusion

This paper investigated carbon-aware RU sleep control in O-RAN under heterogeneous energy supply, where renewable-powered RUs are carbon-neutral and grid-powered RUs incur operational emissions. By formulating the problem as a sequential decision-making task and embedding sustainability objectives into reward design, we systematically evaluated TD3, PPO, and heuristic control across multiple renewable penetration levels, spatial layouts, and traffic conditions. The results reveal a clear operational transition from grid-dominant to renewable-dominant activation as renewable availability increases. In structural terms, TD3 activates on average about one fewer RU than PPO and about 2.5 fewer RUs than the heuristic baseline, indicating a more compact and carbon-aware activation pattern. In energy terms, TD3 consistently outperforms PPO, reducing average network energy consumption by 5–10 W in the 40UE case and 5–8 W in the 80-UE case, while AG1 further achieves 2–4 W lower energy consumption than AG2 in most

moderate-renewable scenarios. In carbon terms, TD3-AG2 achieves 8%–12% lower emissions than PPO-AG2 in low-renewable scenarios, maintains 6%–10% lower emissions under higher renewable penetration, and reaches up to 20% lower emissions than the heuristic baseline under 80 UEs. In addition, AG1 reduces carbon emissions by approximately 12%–28% relative to AG3, with an average reduction of around 18%, confirming that explicit carbon awareness substantially reshapes RU activation behavior. These findings demonstrate that carbon-aware reward design is essential for sustainable O-RAN operation, and that deterministic off-policy learning provides a more effective solution for large-scale RU control under heterogeneous energy conditions.

7

Conclusions and future work

This thesis has investigated intelligent control strategies for improving the energy efficiency, scalability, and sustainability of O-RAN-based radio access networks. By leveraging the programmability and hierarchical architecture of O-RAN, a unified DRL-based framework has been developed to optimize RU activation decisions under dynamic traffic and heterogeneous deployment conditions.

The first part of this work demonstrated that practical energy savings can be achieved through intelligent xApp design within the Near-RT RIC. The proposed control mechanisms, validated using a commercial RIC simulator, achieved up to 50% reduction in power consumption by dynamically adapting RU operational states to traffic demand, while preserving QoS requirements.

To address the inherent complexity of RU activation decisions in dense networks, the problem was formulated as a sequential decision-making task and solved using DRL. The proposed TD3-based approach successfully mitigated the action space explosion problem associated with discrete-action methods, achieving over 50% energy savings and outperforming DQN-based approaches by up to 6% in energy efficiency, while also demonstrating improved convergence stability.

Recognizing the limitations of centralized learning in large-scale deploy-

ments, this thesis further introduced a federated DRL framework aligned with the O-RAN control hierarchy. By distributing learning across multiple regions and aggregating policies at the Non-RT RIC, the proposed approach significantly improved scalability. Simulation results showed up to 43.75% faster convergence and a 37.4% reduction in training energy consumption compared to centralized baselines, highlighting the effectiveness of distributed intelligence in large networks.

Finally, the thesis addressed the critical challenge of environmental sustainability by extending the optimization objective beyond energy efficiency. A carbon-aware DRL framework was proposed to explicitly account for heterogeneous energy sources, enabling differentiated control of renewable-powered and grid-powered RUs. The results demonstrated that carbon-aware policies can achieve up to 20% reduction in emissions compared to heuristic methods, and that joint energy–carbon optimization leads to 12%–28% lower emissions compared to energy-only strategies.

Overall, the contributions of this thesis demonstrate that intelligent control in O-RAN systems must evolve from energy-efficient design toward scalable and sustainability-aware operation. The proposed framework provides a comprehensive solution that integrates efficiency, scalability, and environmental impact, offering a promising direction for future green and intelligent wireless networks.

Future work can further extend this thesis in several directions. First, the mobility and traffic models can be made more realistic by considering time-varying traffic demand, heterogeneous user service requirements, and mobility traces from real network deployments. This would allow the

proposed RU sleep control policies to be evaluated under more diverse and practical operating conditions. Second, the proposed xApp-based and DRL-based control mechanisms can be implemented and tested in real O-RAN testbeds, where practical issues such as signalling delay, measurement uncertainty, control-loop latency, and interoperability with existing RIC platforms can be investigated.

Another important direction is to further improve the scalability and robustness of learning-based RU control. Although the federated reinforcement learning framework developed in this thesis improves scalability compared with centralized learning, future work could investigate asynchronous aggregation, partial model sharing, communication-efficient updates, and robustness against non-identical regional traffic distributions. In addition, future research could explore hybrid control frameworks that combine heuristic rules with AI-based decision-making, allowing the system to benefit from the interpretability and reliability of heuristics while retaining the adaptability of learning-based methods.

Finally, future work should continue to investigate sustainability-aware O-RAN control. The carbon-aware framework developed in this thesis can be extended by incorporating time-varying grid carbon intensity, renewable energy forecasting, battery storage, and electricity price signals. In addition, the energy consumed by AI training and inference should be explicitly included in future optimisation frameworks, so that the net environmental benefit of AI-assisted RAN control can be quantified more accurately. This would support the design of O-RAN control mechanisms that are not only energy-efficient, but also computationally efficient and environmentally sus-

tainable.

References

[1] X. Liang, Q. Wang, A. Al-Tahmeesschi, S. B. Chetty, D. Grace, and H. Ahmadi, "Energy consumption of machine learning enhanced open ran: A comprehensive review," *IEEE Access*, vol. 12, pp. 81889–81910, 2024.

[2] W. Yu, H. Xu, A. Hematian, D. Griffith, and N. Golmie, "Towards energy efficiency in ultra dense networks," in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, IEEE, 2016.

[3] J. S. Thompson, S. Fletcher, V. Friderikos, Y. Gao, L. Hanzo, M. R. Nakhai, T. O'Farrell, and P. D. Wells, "Editorial a decade of green radio and the path to "net zero": A united kingdom perspective," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 657–664, 2022.

[4] J. Wu, S. Zhou, and Z. Niu, "Traffic-aware base station sleeping control and power matching for energy-delay tradeoffs in green cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4196–4209, 2013.

[5] T. Zhou, Y. Fu, D. Qin, X. Li, and C. Li, "Joint user association and bs operation for green communications in ultra-dense heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 2, pp. 2305–2319, 2023.

[6] D. López-Pérez, A. De Domenico, N. Piovesan, G. Xinli, H. Bao, S. Qitao, and M. Debbah, "A survey on 5G radio access network energy efficiency: Massive MIMO, lean carrier design, sleep modes, and machine learning," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 653–697, 2022.

[7] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[8] O-RAN Alliance WG2, "O-RAN non-RT RIC architecture 2.01," October 2022. accessed: 5 Dec 2022.

[9] H. Ju, S. Kim, Y. Kim, and B. Shim, "Energy-efficient ultra-dense network with deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6539–6552, 2022.

[10] D. Sabella, P. Rost, Y. Sheng, E. Pateromichelakis, U. Salim, P. Guitton-Ouhamou, M. Di Girolamo, and G. Giuliani, "RAN as a service: Challenges of designing a flexible RAN architecture in a cloud-based heterogeneous mobile network," in *2013 Future Network & Mobile Summit*, pp. 1–8, IEEE, 2013.

[11] C. Mobile, "C-RAN: The road towards green RAN, white paper, ver. 2.5," *China Mobile Research Institute*, 2011.

[12] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Toward an efficient C-RAN optical fronthaul for the future networks: A tutorial on technologies, requirements, challenges, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 708–769, 2017.

[13] M. A. Marotta, H. Ahmadi, J. Rochol, L. DaSilva, and C. B. Both, "Characterizing the relation between processing power and distance between BBU and RRH in a cloud RAN," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 472–475, 2018.

[14] J. van de Belt, H. Ahmadi, and L. E. Doyle, "Defining and surveying wireless link virtualization and wireless network virtualization," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1603–1627, 2017.

[15] L. Gavrilovska, V. Rakovic, and D. Denkovski, "From cloud RAN to open RAN," *Wireless Personal Communications*, vol. 113, pp. 1523–1539, 2020.

[16] B. Brik, K. Boutiba, and A. Ksentini, "Deep learning for B5G open radio access network: Evolution, survey, case studies, and challenges," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 228–250, 2022.

[17] M. Polese, L. Bonati, S. Doro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[18] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, "Toward next generation open radio access networks–what O-RAN can and cannot do!," *IEEE Network*, 2022.

[19] O-RAN Alliance WG1, "O-RAN Architecture Description," October 2022. accessed: 2 Jan 2023.

[20] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and learning in O-RAN for data-driven NextG cellular networks," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021.

[21] O-RAN Alliance WG6, "Cloud architecture and deployment scenarios for O-RAN virtualized RAN," October 2023. accessed: 12 Oct 2023.

[22] D. Sabella, A. De Domenico, E. Katranaras, M. A. Imran, M. Di Girolamo, U. Salim, M. Lalam, K. Samdanis, and A. Maeder, "Energy efficiency benefits of RAN-as-a-service concept for a cloud-based 5G mobile network infrastructure," *IEEE Access*, vol. 2, pp. 1586–1597, 2014.

[23] T. Pamuklu, M. Erol-Kantarci, and C. Ersoy, "Reinforcement learning based dynamic function splitting in disaggregated green open RANs," in *ICC 2021-IEEE International Conference on Communications*, pp. 1–6, IEEE, 2021.

[24] O-RAN Alliance WG2, "A1 Interface: general aspects and principles 3.0," October 2022. accessed: 12 Dec 2022.

[25] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead," *Computer Networks*, vol. 182, p. 107516, 2020.

[26] O-RAN Alliance WG6, "O-RAN cloud architecture and deployment scenarios for O-RAN virtualized RAN 5.0," June 2021. accessed: 15 Dec 2022.

[27] O-RAN Alliance WG2, "AI/ML workflow description and requirements," October 2021. accessed: 15 Jan 2023.

[28] R. Joda, T. Pamuklu, P. E. Iturria-Rivera, and M. Erol-Kantarci, "Deep reinforcement learning-based joint user association and CU-DU placement in O-RAN," *IEEE Transactions on Network and Service Management*, 2022.

[29] S. Mollahasani, M. Erol-Kantarci, and R. Wilson, "Dynamic CU-DU selection for resource allocation in O-RAN using actor-critic learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2021.

[30] T. Pamuklu, S. Mollahasani, and M. Erol-Kantarci, "Energy-efficient and delay-guaranteed joint resource allocation and DU selection in O-RAN," in *2021 IEEE 4th 5G World Forum (5GWF)*, pp. 99–104, IEEE, 2021.

[31] S. Mollahasani, T. Pamuklu, R. Wilson, and M. Erol-Kantarci, "Energy-aware dynamic DU selection and NF relocation in O-RAN using actor–critic learning," *Sensors*, vol. 22, no. 13, p. 5029, 2022.

[32] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Applications of machine learning in resource management for RAN-slicing in 5G and beyond networks: A survey," *IEEE Access*, vol. 10, pp. 106581–106612, 2022.

[33] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation O-RAN resource slicing for URLLC services using deep reinforcement learning," *IEEE Communications Standards Magazine*, vol. 7, no. 1, pp. 66–73, 2023.

[34] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Federated deep reinforcement learning for resource allocation in O-RAN slicing," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pp. 958–963, IEEE, 2022.

[35] M. Sharara, T. Pamuklu, S. Hoteit, V. Vèque, and M. Erol-Kantarci, "Policy-gradient-based reinforcement learning for computing resources allocation in O-RAN," in *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pp. 229–236, IEEE, 2022.

[36] T. P. Cheng, Nien Fang and M. Erol-Kantarci., "Reinforcement learning based resource allocation for network slices in O-RAN midhaul," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pp. 140–145, IEEE, 2023.

[37] I. Tamim, S. Aleyadeh, and A. Shami, "Intelligent O-RAN traffic steering for URLLC through deep reinforcement learning," *arXiv preprint arXiv:2303.01960*, 2023.

[38] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent traffic steering in beyond 5G Open RAN based on LSTM traffic prediction," *IEEE Transactions on Wireless Communications*, 2023.

[39] H. Erdol, X. Wang, P. Li, J. D. Thomas, R. Piechocki, G. Oikonomou, R. Inacio, A. Ahmad, K. Briggs, and S. Kapoor, "Federated meta-learning for traffic steering in O-RAN," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–7, IEEE, 2022.

[40] A. Lacava, M. Polese, R. Sivaraj, R. Soundrarajan, B. S. Bhati, T. Singh, T. Zugno, F. Cuomo, and T. Melodia, "Programmable and customized intelligence for traffic steering in 5G networks using open RAN architectures," *IEEE Transactions on Mobile Computing*, 2023.

[41] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, *et al.*, "How much energy is needed to run a wireless network?," *IEEE wireless communications*, vol. 18, no. 5, pp. 40–49, 2011.

[42] A. I. Abubakar, O. Onireti, Y. Sambo, L. Zhang, G. Ragesh, and M. A. Imran, "Energy efficiency of open radio access network: A survey," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–7, IEEE, 2023.

[43] Z. Khan, H. Ahmadi, E. Hossain, M. Coupechoux, L. A. DaSilva, and J. J. Lehtomäki, "Carrier aggregation/channel bonding in next generation cellular networks: Methods and challenges," *IEEE network*, vol. 28, no. 6, pp. 34–40, 2014.

[44] G. Yu, Q. Chen, R. Yin, H. Zhang, and G. Y. Li, "Joint downlink and uplink resource allocation for energy-efficient carrier aggregation," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3207–3218, 2015.

[45] R. Singh, C. Hasan, X. Foukas, M. Fiore, M. K. Marina, and Y. Wang, "Energy-efficient orchestration of metro-scale 5G radio access networks," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10, IEEE, 2021.

[46] H. B. Nafea, M. M. Sallam, and F. W. Zaki, "Study of DRX sleep mode performance on virtual base station energy saving in 5G networks," *Wireless Personal Communications*, vol. 118, pp. 3251–3270, 2021.

[47] T. Zhao, J. Wu, S. Zhou, and Z. Niu, "Energy-delay tradeoffs of virtual base stations with a computational-resource-aware energy consumption model," in *2014 IEEE International Conference on Communication Systems*, pp. 26–30, IEEE, 2014.

[48] S. Maxenti, S. D'Oro, L. Bonati, M. Polese, A. Capone, and T. Melodia, "ScalO-RAN: Energy-aware network intelligence scaling in Open RAN," *arXiv preprint arXiv:2312.05096*, 2023.

[49] J. Gong, J. S. Thompson, S. Zhou, and Z. Niu, "Base station sleeping and resource allocation in renewable energy powered cellular networks," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3801–3813, 2014.

[50] Y. Sun, H. Zhou, K. Yu, Y. Xu, B. Qian, and L. X. Cai, "Flexible base station sleeping and resource allocation for green uplink fully-decoupled ran," *IEEE Transactions on Wireless Communications*, 2025.

[51] J. Wu, Y. Li, H. Zhuang, Z. Pan, G. Wang, and Y. Xian, "Smdp-based sleep policy for base stations in heterogeneous cellular networks," *Digital Communications and Networks*, vol. 7, no. 1, pp. 120–130, 2021.

[52] X. Liang, A. Al-Tahmeesschi, Q. Wang, S. Chetty, C. Sun, and H. Ahmadi, "Enhancing energy efficiency in O-RAN through intelligent xapps deployment," *arXiv preprint arXiv:2405.10116*, 2024.

[53] Q. Wang, S. Chetty, A. Al-Tahmeesschi, X. Liang, Y. Chu, and H. Ahmadi, "Energy saving in 6g o-ran using dqn-based xapp," *arXiv preprint arXiv:2409.15098*, 2024.

[54] G. Jang, N. Kim, T. Ha, C. Lee, and S. Cho, "Base station switching and sleep mode optimization with lstm-based user prediction," *IEEE Access*, vol. 8, pp. 222711–222723, 2020.

[55] J. Ye and Y.-J. A. Zhang, "Drag: Deep reinforcement learning based base station activation in heterogeneous networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2076–2087, 2019.

[56] Q. Wu, X. Chen, Z. Zhou, L. Chen, and J. Zhang, "Deep reinforcement learning with spatio-temporal traffic forecasting for data-driven base station sleep control," *IEEE/ACM transactions on networking*, vol. 29, no. 2, pp. 935–948, 2021.

[57] J. Gan, H. Kou, G. Yang, H. Zhang, Z. Cao, and W. Xu, "Joint sleep control and energy sharing strategy with deep reinforcement learning in green ultra-dense networks," *IEEE Transactions on Green Communications and Networking*, 2025.

[58] S. Sun, C. Huang, G. Chen, P. Xiao, and R. Tafazolli, "Deep learning-based traffic-aware base station sleep mode and cell zooming strategy in ris-aided multi-cell networks," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[59] M. Masoudi, E. Soroush, J. Zander, and C. Cavdar, "Digital twin assisted risk-aware sleep mode management using deep q-networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1224–1239, 2022.

[60] Y. Zhen, L. Tao, D. Wu, T. Tang, and R. Wang, "Energy-saving control strategy for ultra-dense network base stations based on multi-agent reinforcement learning," *Digital Communications and Networks*, 2024.

[61] Y. Zhou, Y. Shi, H. Zhou, J. Wang, L. Fu, and Y. Yang, "Toward scalable wireless federated learning: Challenges and solutions," *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 10–16, 2023.

[62] B. Zhao, Q. Cui, W. Ni, X. Li, and S. Liang, "Multi-layer collaborative federated learning architecture for 6g open ran," *Wireless Networks*, vol. 31, no. 2, pp. 1377–1390, 2025.

[63] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang, "Federated reinforcement learning with environment heterogeneity," in *International Conference on Artificial Intelligence and Statistics*, pp. 18–37, PMLR, 2022.

[64] T. Pan, X. Wu, and X. Li, "Dynamic multi-sleeping control with diverse quality-of-service requirements in sixth-generation networks using federated learning," *Electronics*, vol. 13, no. 3, p. 549, 2024.

[65] VIAVI Solutions Inc., "Viavi network simulation and test solutions," 2024.

[66] E. Björnson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Optimal design of energy-efficient multi-user MIMO systems: Is massive MIMO the answer?," *IEEE Transactions on wireless communications*, vol. 14, no. 6, pp. 3059–3075, 2015.

[67] ETSI, "TS132425-V15.1.0-LTE," June 2018. accessed: 2024-03-25.

[68] "Study on channel model for frequencies from 0.5 to 100 GHz ." `https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf`. Accessed: 2024-03-25.

[69] G. Shani, D. Heckerman, and R. I. Brafman, "An mdp-based recommender system," *Journal of machine Learning research*, vol. 6, no. Sep, pp. 1265–1295, 2005.

[70] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[71] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[72] Y. Liu, H. Zhou, Y. Deng, and A. Nallanathan, "Channel access optimization in unlicensed spectrum for downlink urllc: Centralized and federated drl approaches," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 7, pp. 2208–2222, 2023.

[73] A. Ndikumana, K. K. Nguyen, and M. Cheriet, "Federated learning assisted deep q-learning for joint task offloading and fronthaul segment routing in open ran," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3261–3273, 2023.

[74] J. Wang, P. Chen, J. Wang, and B. Yang, "A hierarchical federated learning paradigm in o-ran for resource-constrained iot devices," in *ICC 2024-IEEE International Conference on Communications*, pp. 2555–2560, IEEE, 2024.

[75] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[76] A. K. Singh and K. K. Nguyen, "Communication efficient compressed and accelerated federated learning in open ran intelligent controllers," *IEEE/ACM Transactions on Networking*, 2024.

[77] 3GPP, "Study on channel model for frequencies from 0.5 to 100 ghz," Technical Report TR 138 901, ETSI, 2017.